AWARD NUMBER: W81XWH-13-2-0086

TITLE: Optimizing Soft Tissue Management and Spacer Design in Segmental Bone Defects

PRINCIPAL INVESTIGATORS: Dr. George F. Muschler

CONTRACTING ORGANIZATION:
Cleveland Clinic Foundation
Cleveland, OH44195

REPORT DATE: **December 2016**

TYPE OF REPORT: **Final**

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. REPORT DATE	**2. REPORT TYPE**	**3. DATES COVERED**
December 2016	Final Report	30 SEP 2013 to 29 SEP 2016

4. TITLE AND SUBTITLE	**5a. CONTRACT NUMBER**
Optimizing Soft Tissue Management and Spacer Design in Segmental Bone Defects	**5b. GRANT NUMBER** W81XWH-13-2-0086
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	**5d. PROJECT NUMBER**
Dr. George F. Muschler	**5e. TASK NUMBER**
E-Mail(s): muschlg@ccf.org	**5f. WORK UNIT NUMBER**

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT**
The Cleveland Clinic 9500 Euclid Avenue Cleveland, OH 44195	

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
U.S. Army Medical Research and Materiel Command Fort Detrick, Maryland 21702-5012	**11. SPONSOR/MONITOR'S NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This proposal addresses the treatment of segmental bone defects and methods that can be used to manipulate the Masquelet induced membrane to create a graft bed that optimizes bone regeneration. The effect of surgical management of the IM demonstrated a significant benefit of scraping to remove the inner layer of the IM (p=0.041). In contrast, modifying the texture on the PMMA spacer to double the surface area of the IM did not result in improved bone regeneration. This finding can be immediately translated into clinical practice. This study also highlights that the Chronic Caprine Tibial Defect Model is sufficiently sensitive to detect variations and represents a particularly well suitable relevant large animal model to advance the field of clinical bone regeneration using cellular therapies, and optimization of clinically relevant methods for harvest and processing of CTP sources to enhance the concentration and prevalence CTPs in the site of bone regeneration.

15. SUBJECT TERMS
Optimization of the Masquelet induced membrane to improve bone healing.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION**	**18. NUMBER**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	6	**19b. TELEPHONE NUMBER** *(include area code)*

Table of Contents

Page

1. INTRODUCTION:

High energy extremity fractures with soft tissue involvement affect thousands of U.S. Service members each year [1]. These injuries are complex in nature, typically involving large regions of bone damage or loss and often associated with multiple tissue loss such as skin, bone, muscle, cartilage, vascularity, nerves. In some cases, these injuries also are complicated by infection. Despite substantial advances in the availability of bone graft substitute materials in recent years, these large bone defects remained an unsolved clinical challenge. Additionally, although a massive amount of research on bone defects has been done, with astoundingly good results in animal models, results are mediocre at best in the real clinical situation. This research specifically addresses the treatment of segmental bone defects and methods that can be used to optimize the Masquelet induced membrane [2-6] technique to create a superior graft bed for bone regeneration. Using the Chronic Caprine Tibial Defect (CCTD) model, this study specifically aims to assess the effects of surgical technique and spacer design in optimizing the biology of the "Induced Membrane" (IM), and to define the characteristics of an optimal membrane with respect to the parameters of cell composition, histology, and gene expression. Optimizing the Masquelet IM method and characterization of the biological features of the IM has significant potential to enhance the clinical care of wounded warriors who require bone regeneration procedures and to guide the ongoing development of advanced methods for bone regeneration. This program includes a nested development program for Dr. Jean-Claude D'Alleyrand, a junior orthopaedic surgeon at the Walter Reed National Military Medical Center, to nurture his interest in translational and clinical research by fostering relationships with established, independent clinician scientists with proven track records.

The Specific Aims of this proposal were as follows:

Aim 1 – Test the hypothesis that removal of the thin inner surface of the IM created around a smooth polymethylmethacrylate (PMMA) spacer will enhance bone regeneration.

Aim 2 - Test the hypothesis that a textured PMMA spacer will enhance bone regeneration compared to a smooth PMMA spacer.

Aim 3 – Characterize the histological, biochemical, cellular and gene expression features of the IM and define the features that best predict the magnitude of bone regeneration following an Autogenous cancellous Bone graft (ACBG).

The broad objective of our team is to accelerate the rate at which clinically significant questions related to surgical management of the "induced membrane", advanced spacer design, and advanced bone regeneration strategies can be translated into clinical practice to improve the care of wounded warriors and civilians.

2. KEYWORDS:

KEYWORDS: segmental bone defect, caprine, tibia, animal model, chronic, bone graft, Masquelet technique, induced membrane, bone regeneration, surgical technique, spacer.

3. OVERALL PROJECT SUMMARY:

2.1 Overall Summary for Aim 1 and Aim 2:

Aim 1: *Test the hypothesis that removal of the thin inner surface of the IM created around a smooth polymethylmethacrylate (PMMA) spacer will enhance bone regeneration.*

Aim 2: *Test the hypothesis that a textured PMMA spacer will enhance bone regeneration compared to a smooth PMMA spacer.*

Protocol

A total of 32 animals were used in the study. Thirty-two skeletally mature female goats, age 5 ± 1 years (mean ± SD) weighing 50 ± 4 kg underwent the CCTD procedure and were randomly assigned to four groups (8 animals/group) using a 2 x 2 test matrix (factorial design) (Figure 1). These groups included: (1) smooth spacer and intact IM; (2) smooth spacer and scraped IM; (3) textured spacer and intact IM; and (4) textured spacer and scraped IM. Study animals were cared for in accordance with the principles of the Guide for the Care and Use of Laboratory Animals after approval from the Cleveland Clinic Institutional Animal Care and Use Committee (IACUC # 2013-1021) and the Animal Care and Use Review Office of US Army Medical Research and Materiel Command (OR # 120082).

The textured spacer had a surface area that doubled the surface area of a smooth spacer.

Aim 1 (16 goats):
- Group 1 –Smooth spacer and not scraped IM
- Group 2 – Smooth spacer and scraped IM

Aim 2 (16 goats):
- Group 1 – Textured spacer and not scraped IM
- Group 2 – Textured spacer and scraped IM

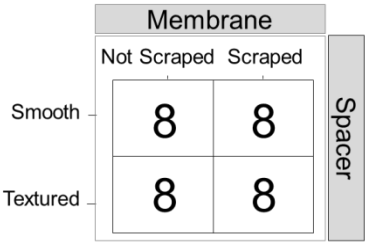


Figure 1 – Test matrix used

Surgical Protocol

Each animal undergoes two surgeries defined here as: 1) the "pre-procedure" to create the tibia defect and the IM and 2) the "treatment" (4 weeks after "pre-procedure"). Autogenous cancellous bone graft (ACBG) is placed into the IM that is scraped for half of the goats and not scraped in the other half.

The "Pre-Procedure" (Figure 2 A & D) is comprised of the following essential features:

1. Make a medial skin incision and excise a 5-cm segment of tibial diaphysis and periosteum.
2. Excise an additional 2 cm of periosteum on the proximal and distal bone segments.
3. Debride 10 grams of tibialis anterior and gastrocnemius muscles.
4. Place an interlocking intramedullary nail using a custom spacer to maintain 5-cm defect length.
5. Place a pre-molded 5 cm long x 2 cm diameter PMMA spacer around the nail in the defect.
6. Irrigate the wound with normal (0.9 %) saline and close the wound.

The "Treatment" (Figure 2B,C, E & F) is performed 4 weeks after the "Pre-Procedure" and is comprised of the following steps:

1. Collect ACBG from sternebrae. 12 cc of cancellous graft is needed for each defect; all graft available is collected from sternebra 4 or 5, approaching other sternebra only as needed.
2. Open the previous skin incision on the medial aspect of the tibia.
3. Open the IM surrounding the PMMA spacer using a "bomb bay door opening".
4. Remove the spacer without damaging the membrane or nail.
5. Collect appropriate IM samples as defined below (see section b).
6. In Aim 1, the inner layer of the IM was scraped away in one half of the goats before grafting and the graft was placed in the intact IM in the remaining goats.
7. Close the IM with 3-0 nylon to provide an intrinsic marker and close the remaining tissues.

The procedures after treatment include:

- Orthogonal radiographs (anterior-posterior (AP) and mediolateral (ML) projections) of tibias every 4 weeks
- Physical examination including lameness grading daily, then biweekly starting 2 weeks after "Treatment" surgery.

Euthanasia is performed 12 weeks after "Treatment" surgery at which time tibias are harvested and fixed in 10% formalin. Micro CT and histologic analyses of regenerate tissue are then performed.

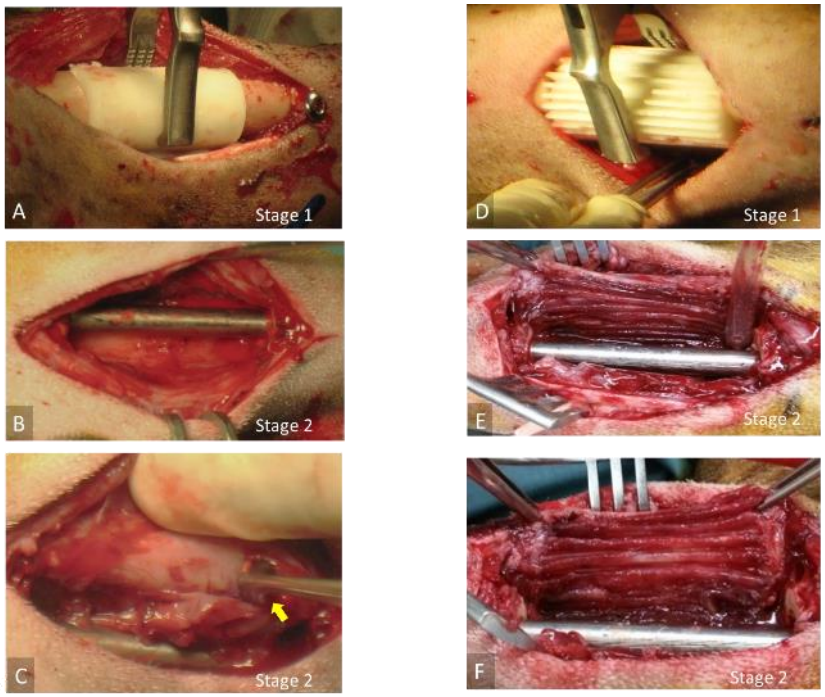


Figure 2 – Surgery pictures showing placement of a smooth spacer (A) and textured spacer (D) in the defect site during the "pre-procedure" (Stage 1), the induced membrane four weeks later during the "treatment procedure" (Stage 2) after smooth spacer (B) or textured spacer (E) removal, and after scraping the smooth (C) and textured (F) IM.

MicroCT processing and analysis

High-resolution microCT was used as the primary outcome measure to characterize the amount and distribution of bone within each defect. Prior to scanning, the nail and screws were carefully removed and replaced with a radiolucent rod and cross-pins to maintain the original length and axial and rotational alignment. After centering in the scanner (Inveon microCT scanner, Siemens Medical Solutions, Knoxville, TN), 441 projection images were acquired at 0.5° increments at 39-µm voxel resolution (80 kVp; 500 µA). Data were reconstructed into 3D volumes using a modified tent-Feldkamp algorithm and calibrated to mg/cm3 of HA using an air/water/hydroxyapatite phantom scanned under the same conditions. Bone threshold was set at 1300 mg HA/cm^3 (747 HU). The analyzed volume of interest (VOI) included the 5 cm defect region plus 1 cm of bone proximal and distal to the defect. Each specimen was analyzed with a segmentation software developed in-house, in which a 3-dimensional cylindrical "defect template" volume 45mm in diameter and 70 mm in length, was manually positioned to define the boundaries of the VOI. Primary outcome was defined as total bone volume (tBV) in the central 2.5 cm region of the defect, which is the most challenging region to heal. Secondary outcomes were radial percent bone volume (%BV), a summation of circumferential bone volume over the length of the 5cm defect, and angle-moment of inertia in the defect site. Summary data of the pattern and extent of mineralization in the defects in each group were illustrated by projecting %BV versus radial position using a 2D color map ranging from 0% (purple) to 60% (red) (Figure 3). The x-axis indicated distance from the center of the medullary canal to the periosteal surface (range = 0-29 mm). The y-axis represented the position in the long axis of the bone (range 0 - 70 mm) including the 5 cm defect plus 1 cm proximal and distal. Mean Angle-moment of inertia for each group was also plotted using a 2D color plot color map ranging 0 (dark blue) to 7000 (red) in Hounsfield units (HU)*mm^2 plotted about the bone circumference in the x-axis and the long bone axis position in the y-axis.

Radiographic Analysis

Fluoroscopic imaging of the tibiae, anterior-posterior (AP) and medio-lateral (ML) projections, were performed after: the spacer procedure (week 0), the graft procedure (week 4), follow-up (week 8).

Radiographs were obtained after euthanasia (after soft tissues were dissected) 12 weeks after the grafting procedure. The resulting images were ranked from 1 (greatest bone healing) to 20 (no healing) by two independent investigators (senior orthopaedic surgeon and junior orthopaedic surgeon) who were blinded to treatment allocation. The highest-ranking sites revealed bony bridging (bone extending the length of the defect with no discontinuity) of all four cortices, followed by bony bridging of three, two, one, or none of the cortices. Among samples that had the same number of bony bridges, the ranking was based on the subjective amount of bone observed.

Histology Analysis

Following microCT data collection, histology and histomorphometry were performed. Each fixed tibia was immersed in dilute HCl (~1 week) and then transferred to a 10% EDTA solution until completely decalcified. Decalcified specimens were embedded in paraffin, trimmed to 7 cm to include the 5 cm defect and 1 cm at either end. The specimens were compared with the corresponding Faxitron radiographs to ensure that the correct tissue area was examined. The specimens were then divided into four 1.75 cm long segments that were then cut in the craniocaudal direction on the mid-sagittal plane to yield medial and lateral halves. Medial halves (each including the anterior and posterior cortices) were processed in paraffin (four blocks/specimen), sectioned at 5-µm, and stained with hematoxylin and eosin (H&E) and Masson's trichrome. For the histomorphometric analysis (Figure 3 D & E), total area of bone in the sections from the two central 1.75 cm blocks (total of 3.5 cm) was measured by tracing the perimeter of each individual focus/area of bone tissue in the section in both anterior and posterior cortices (mm^2) using SPOT basic histomorphometry software (SPOT Imaging, Sterling Heights, MI) and summing the results for each animal.

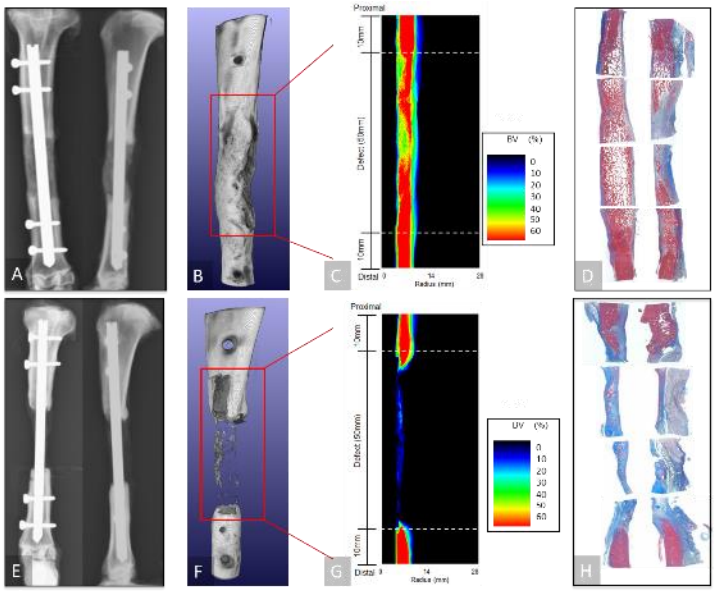


Figure 3 - Example of postmortem radiographs (AP and ML projections), MicroCT scan 3D reconstruction, % BV plot versus summation of radial position and Masson's Trichrome stained histology slides for two goats showing robust new bone (pink tissue) formation (A, B, C, D) or minimal new bone formation (E, F, G, H).

Statistical Analysis

All statistical tests were performed using general linear models using one-sided tests at a significance level of 0.05 with SAS 9.4 software (SAS Institute, Cary, NC). The response variable was total bone volume (tBV in mm^3) calculated from the microCT scans in the central 2.5 cm of the defect and the treatment factors were scraped or non-scraped IM and textured or smooth spacers. All data were expressed as mean ± standard error.

Results:

Three goats were excluded due to complications. Two belonged to the intact/smooth group and were excluded due to Staphylococcus epidermidis infection (n=1) or caseous lymphadenitis (n=1); one goat in the scraped/smooth group was excluded due to anesthetic complication during stage 2 surgery (n=1).

Bone formation assessed by MicroCT

Mean percent bone volume in Figure 4A illustrates that there is greater bone formation near the osteotomy sites. The angle-moment of inertia plots illustrate that bone tended to form along the posterior and medial aspect of the defect with little bone formation laterally in the IM scraped group while in the intact IM group, new bone formation was found posteriorly (Figure 4B). Textured and smooth spacer groups had comparable new bone formation; with bone preferentially forming posteriorly in both groups as shown in the angle moment plots.

Mean radial tBV in the central 2.5 cm of the defect was significantly greater in the IM scraped group (mean +/- standard error = 1861.0 ± 351.3 mm^3) vs. non-scraped IM (930.3 ± 390.3 mm^3) ($p = 0.041$) (Figure 5). However, there was no trend or significant difference between the smooth or textured spacer groups ($p = 0.475$) (Figure 5). The mean tBV in the central 2.5 cm of the defects was 1413.8 ± 363.3 mm^3 for the smooth membrane group and 1377.5 ± 377.5 mm^3 for the textured membrane group. These data are graphically presented using a heat map.

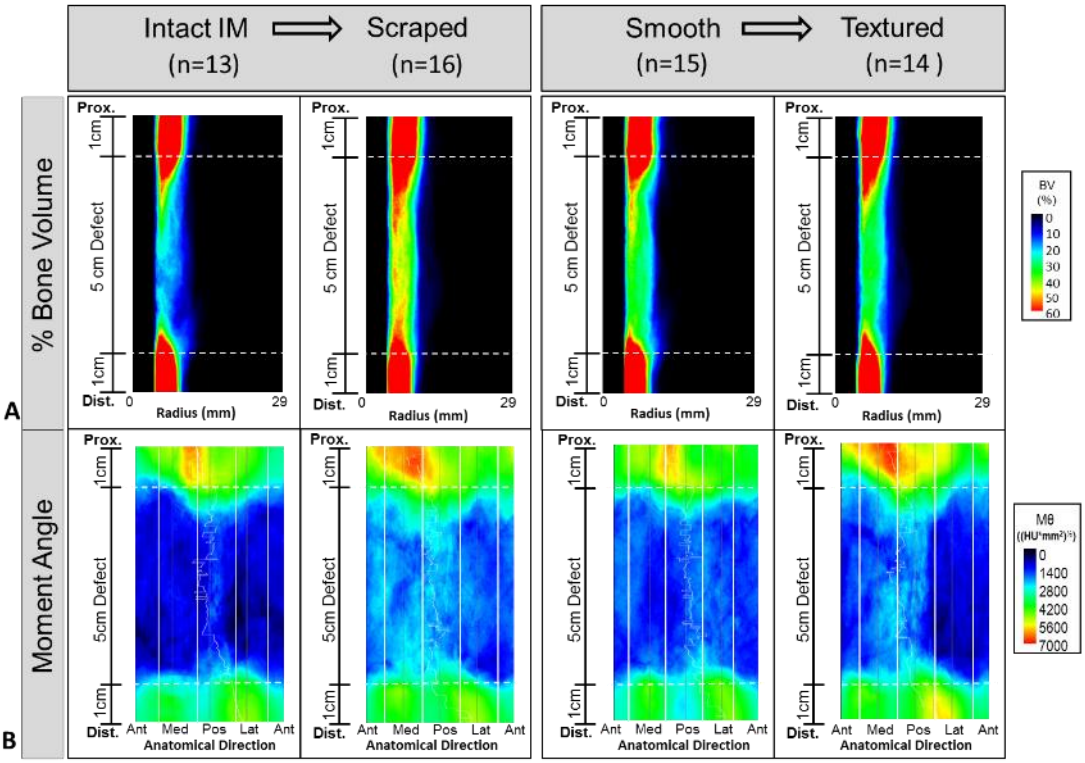


Figure 4 – MicroCT plots illustrating the differences in bone formation and distribution. A) A radial color plot illustrates the probability of bone formation as the summation of the radial location relative to vertical position in the defect. B) A Moment-Angle plot illustrates that most bone formed on the posterior aspect of this defect. The proximal and distal borders of the defect are marked by dashed lines.

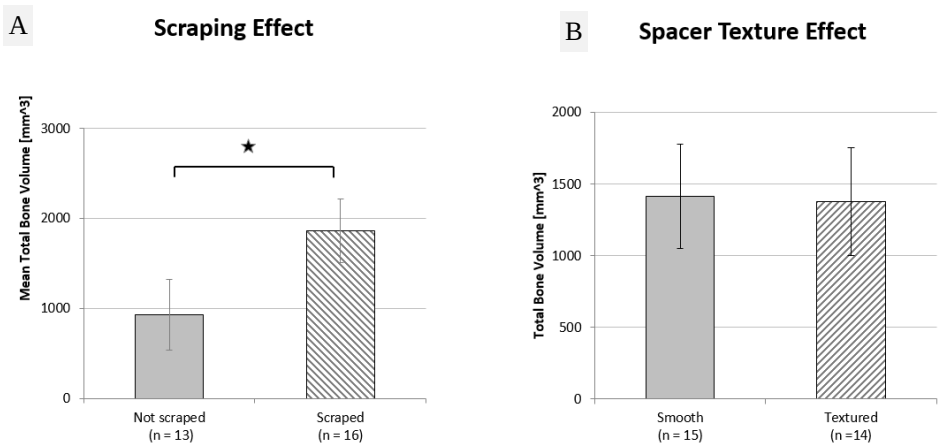


Figure 5 – Total bone volume plots, based on microCT analyses, of the central 2.5 cm of the defect and the entire 5 cm defect illustrating the effect of scraping the IM (A) and the effect of the spacer texture (B).

Postmortem radiograph ranking

Overall, the scraped IM group ranked better than the non-scraped IM group (Figure 6). Ten of the top twelve radiographs belonged to the scraped IM group. The radiographic ranking did not reveal any differences between smooth and textured spacer groups.

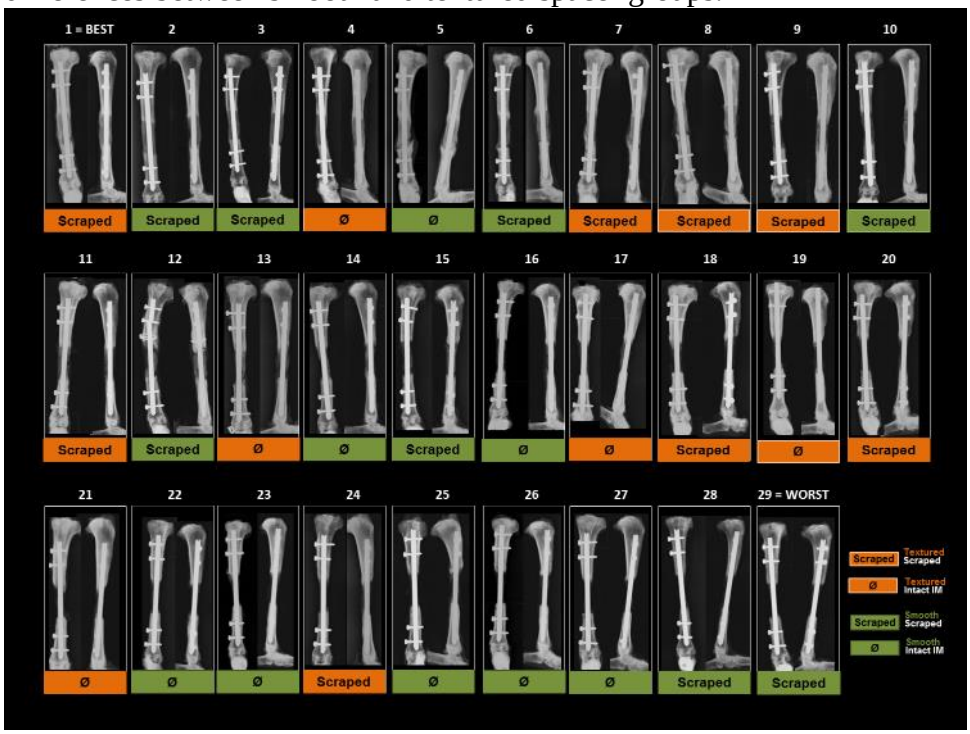


Figure 6 – Postmortem radiographs (AP and ML views) of the tibia from each goat used for ranking. Higher rank number indicates greater bone formation in the tibial defect (1 = complete bone healing to 29 = no healing).

Histology

Histological assessment of the goat tibia samples ((Figure 7) identified no evidence of inflammation in any of the sections 12 weeks after grafting. The new bone that was present in the best healed tibias was similar in quality and was mainly composed of cancellous woven bone. Defects containing little or no bone were filled primarily with fibrous connective tissue. The most robust bone regeneration occurred in the scraped IM group (mean total bone area = 244.1 ± 67.8 mm^2) compared to the intact IM group (177.0 ± 74.3 mm^2), which is consistent with the microCT data. The histomorphometry data also showed that textured spacer group (mean total bone area = 296.9 ± 94.3 mm^2) had a larger area of

bone in the defect than smooth spacer group (136.7 ± 30.4 mm^2). No statistically significant difference was found with histomorphometry assessment between groups.

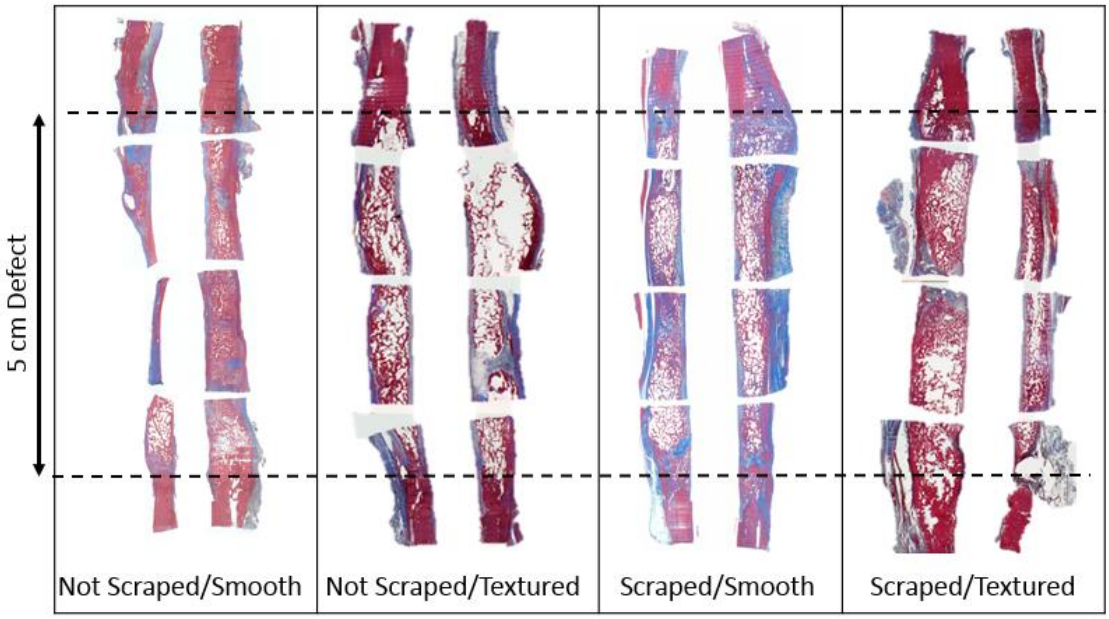


Figure 7 – Examples of histological sections stained with Masson trichrome stain. These are the sections from the tibia from each treatment group that had the highest bone area. In these sections, the bone is red and fibrous connective tissue is blue. There were no obvious differences in the quality of the bone tissue that was formed. Robust bone formation primarily was composed of cancellous woven bone. Defects containing little or no bone were filled primarily with fibrous connective tissue.

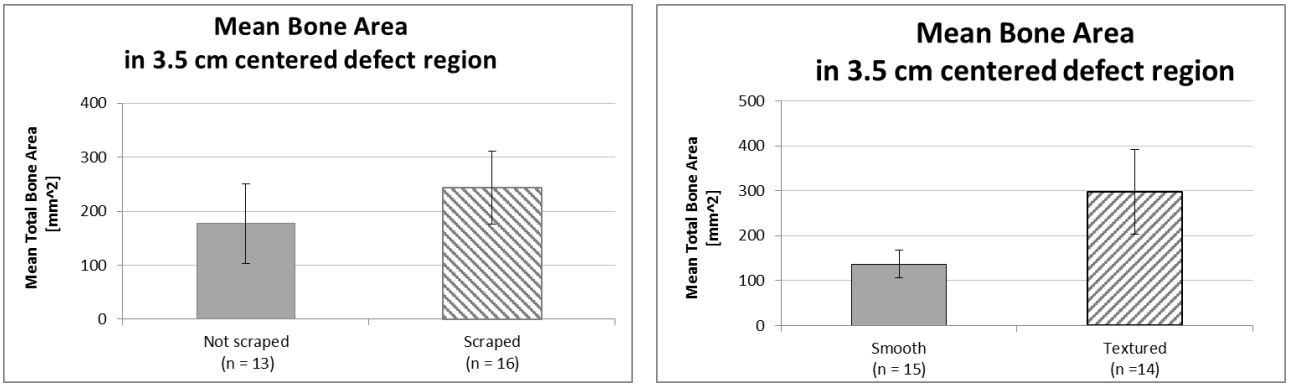


Figure 8 – Histomorphometry Data showing no significant differences in bone area vs. treatment.

2.2 Overall Summary for Aim 3:

Aim 3: *Characterize the histological, biochemical, cellular and gene expression features of the IM and define the features that best predict the magnitude of bone regeneration following an ACBG.*

a) Analysis of Autograft cancellous bone graft (ACBG) and Bone Marrow harvested from sternum

Protocols:

ACBG Histology Protocol: All samples were shipped to the laboratory of Dr. Carlson at University of Minnesota. A 1-cm3 ACBG sample collected for histology analysis was stained with H&E, examined by a board-certified pathologist, and imaged using 1X and 10X objectives. Each sample was characterized by histomorphometry to define areas of hematopoietic marrow, bone, adipose tissue,

blood vessels, fibrous tissue, and void spaces. All values are expressed as Mean area ± Standard Deviation in mm^2.

Bone Marrow and ACBG Cell/Colony Protocol: All samples were processed in the laboratory of Dr. Muschler at Cleveland Clinic. Marrow was aspirated (2 ml) into a 10-cc syringe containing 1 cc of 0.9% saline containing 1000 U/ml heparin. ACBG (total volume of 10-14 cc) was then collected from the 5th sternebrae (distal), with additional collections from the 4^{th} or 6^{th} sternebra, as needed. Two cell populations were collected from the ACBG sample. Cells that could be mechanically disassociated from the ACBG were defined as marrow space cells (MS) and cells that were disassociated only after enzymatic digestion using Collagenase I and dispase, were defined as trabecular surface cells (TS). Nucleated cells were counted for each sample. The biological performance of CTP-Os from each sample was assessed using a quantitative colony forming unit assay.

Results:

The ACBG morphology of all goats was similar, with little variation from sample to sample. ACBG histological samples contained abundant bone trabeculae and bone marrow; bone marrow was moderately cellular in all sections. Histomorphometric analysis of the ACBG samples revealed the following percentages of total tissue area: hematopoietic marrow = 73.8 ± 14.5 %; bone = 19.6 ± 7.7 %; and hyaline cartilage = 5.8 ± 14.4 %. Blood vessels, fibrous tissue and void spaces were either absent or occupied <1% of the tissue area.

Large variation was seen among animals with respect to cell and CTP-O yield from MS, TS and BMA samples, and in the relative distribution of CTP-Os between MS and TS fractions within ACBG samples. Overall, the cellularity in the MS fraction was 2 fold higher than the BMA sample and almost 3 fold higher than in the TS fraction (see Table 1). However, the CTP-O prevalence in the TS fraction was approximately 6.5 fold higher than in the MS fraction; and approximately 9 fold higher than BMA (see Table 1). We also noted that the total cell count after 6 day in culture was higher in the MS fraction than in the TS fraction, which might indicates that a larger population of cells in the MS fraction were single cells, not associated with colony formation.

The yield of cells and CTP-Os, and the associated histology of tissue from the caprine sternum are comparable to human iliac crest; thus, this site appears to be suitable for harvest of bone or marrow for this animal model.

These data demonstrate that the overall concentration of CTPs in a BMA is approximately 5 fold lower than the combined concentration of CTPs in ACBG tissue sample of comparable volume. This decrease in concentration is due to inefficiency in extraction of the CTP population using simple aspiration and the dilution of the ACBG-derived CTPs with peripheral blood. These data conversely demonstrate that bone marrow processing methods that would increase the concentration of CTPs by just 5 fold would enable a surgeon to transplant a population of cells containing a comparable concentration of CTPs to ACBG. These data are exactly analogous to findings demonstrated when comparing human iliac crest core samples with samples of BMA from the same iliac crest [8]. These findings suggest that the CCTD model is well suited to further assessment of clinically relevant options to improve the outcome bone regeneration procedures using bone marrow processing.

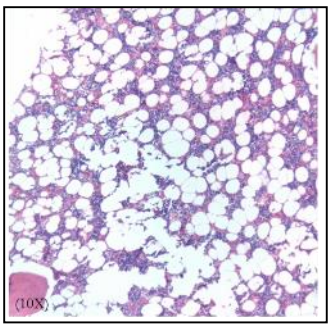


Figure 9- Representative histological image of a caprine sternum ACBG graft showing foci of cancellous bone accompanied by abundant hematopoietic cells (10X; H&E).

	Sternum MS fraction N=32	Sternum TS fraction N=32	Sternum BMA N=32
Cells (Millions/gram or cc)	163.5 ± 11.7 (139.6,187.4)	56.5 ± 5.9 (44.4, 68.6)	79.4 ± 9.3 (60.4, 98.3)
CTP-O Prevalence (CTPs per Million cells plated)	56.3 ± 8.6 (38.7,73.8)	367.7 ± 70.9 (223.04,512.4)	39.4 ± 5.9 (27.4,51.5)
[CTP-O] (CTP per gram tissue or aspirate)	8019.0 ± 1393.1 (5177.8, 10860.1)	14167.8 ± 3309.0 (7419.0, 20916.6)	4133.0 ± 923.4 (2249.7, 6016.3)

Table 1. Cell and CTP Yield of MS fraction, TS fraction, BMA. All values are Mean ±SD (Confidence Intervals).

b) Analysis of Induced membrane sample

Protocol:

The protocol for sampling the induced membrane at Treatment surgery is illustrated in Figure 10. Data collected from each sample include weight and thickness. Specific segments are analyzed for cells and osteogenic connective tissue progenitors (CTP-Os) ("C"), histology ("H"), and gene expression ("GE"). In the case of "C" and "GE" samples, each segment is divided into an "inner layer" (thin, friable and vascular, immediately adjacent to the spacer) and an "outer" layer (thicker, fibrous, mechanically robust, and less vascular. Samples for cell and CTP analysis are minced and digested in collagenase I and dispase. A standard assay for CTP-Os is performed. Two samples H2 and H7 were fixed in 10% neutral buffered formalin and transferred to 70% ethanol at 48 hours then shipped to the laboratory of Dr. Carlson at the University of Minnesota for processing. Two samples GE3 and GE6 were collected, stored in RNA later at 4^0C then shipped to Dr. Davis at the NMRC for gene expression analysis.

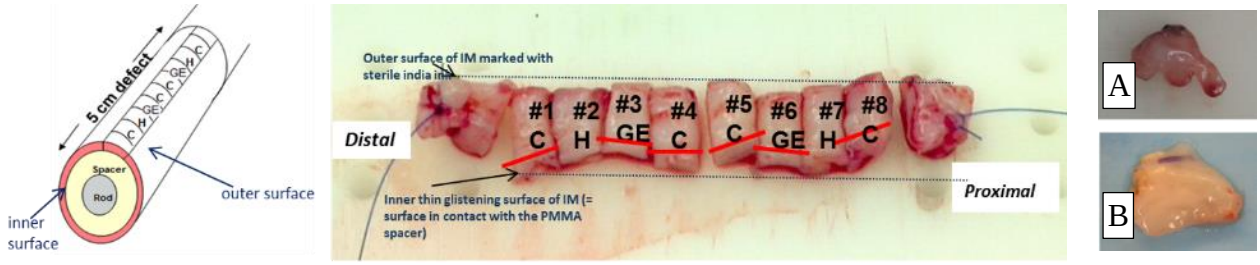


Figure 10 –Induced Membrane sample (7 mm width x 5cm long) excised at the "treatment" surgery. Each strip of IM is sectioned into inner and outer portions for analysis of cells and gene expression (GE3, GE6). The histology sections are kept intact. A) Example picture of a textured induced membrane section; B) Example picture of a smooth induced membrane section picture

Induced membrane histology protocol: All samples were shipped to the laboratory of Dr. Carlson at University of Minnesota. All IM samples were stained with H&E and trichrome stains, examined by a

board-certified pathologist, and imaged using a 1X and 10X objective. A grading scheme for the induced membrane samples has been developed to define overall thickness, cellularity, fibrosis, and vascularity. Grading scheme for Induced Membranes is defined below:

1) If sample contains clearly defined inner and outer zones, measure total thickness in three equidistant sites and express as mean thickness; do not measure samples that appear to have been cut tangentially.

2) Grade the following, on a 0-4 scale:

Fibrosis (on trichrome sections)

- 0 = none
- 1 = < 10% of tissue area is trichrome positive
- 2 = 10-25% of tissue area is trichrome positive
- 3 = >25-75% of tissue area is trichrome positive
- 4 = >75% of tissue area is trichrome positive

Vascularity

- 0 = none
- 1 = minimal (very few vessels present, <5)
- 2 = mild (low numbers of vessels present diffusely throughout the section, 5-10)
- 3 = moderate (contains areas of increased numbers of vessels 9vessel area > tissue area)
- 4 = severe (majority of section contains vessel area that is greater than tissue area)

Cellularity

- 0 = No inflammatory cells
- 1 = Inflammatory cells confined to inner 10% of membrane thickness
- 2 = Inflammatory cells present in 10-25% of membrane thickness
- 3 = Inflammatory cells present in 25-75% of membrane thickness
- 4 = Inflammatory cells present in >75% of membrane thickness

Induced membrane cell/CTP analysis protocol: All samples were processed in the laboratory of Dr. Muschler at Cleveland Clinic. Using a customized parallel knife instrument, a "strip sample" of the IM was harvested from each animal and divided into subsamples (see Fig 1). "C" segments were analyzed for cells and osteogenic connective tissue progenitors (CTP). Each segment was divided into an "inner layer" (zone immediately adjacent to the spacer) and an "outer" layer. All inner and outer segments from C1, C4, C5 and C8 were pooled together and analyzed as two separate groups. "C" Samples were minced and incubated in 10 ml of prepared digest (111 U/ml collagenase I + 24U/ml dispase) for 1.5 hours at 37^{o} C. Complete media was added to stop digestion, the sample was filtered through a cell strainer, and a cell count was obtained. Cells were cultured at 50 x 10^3 cells/chamber for the membranes obtained with smooth spacer (set 1) and 25 x 10^3 cells/chamber for the membranes obtained with textured spacer (set 2) in osteogenic media, harvested on day 6. Day 6 cultures were fixed with 1:1 acetone:methanol for 10 min and stained for nuclei (4,6-diamidino-2-phenylindole [DAPI]) and alkaline phosphatase (AP) as a marker for the preosteoblastic activity. Chambers were scanned and analyzed using Colonyze™ software for colony assessment [7].

Induced membrane gene expression analysis protocol:

Induced membranes samples were obtained after a smooth (data set 1) or textured (data set 2) spacer. Sample sections (GE3, GE6)were collected along the tibia shaft either proximally ("6") or distally ("3") from the hip. Both an outer (o) and inner (i) layer of the induced membrane was dissected along with tibial muscle for control tissue and immediately stored in RNAlater at 4°C. Samples were shipped to NMRC (Dr. Davis/Forsberg) for analysis.

Total RNA is extracted using the RNeasy Plus Universal Kit (Qiagen, Gathersburg,MD). mRNA quantity and quality are assessed by measuring the $A_{260/280}$ and $A_{260/230}$ ratio with a Nanodrop Spectrophotometer (NanoDrop Technologies Inc., Wilmington,DE) and the 28S/18S rRNA ratio and RNA integrity number (RIN) using an Agilent 2100 BioAnalyzer (Agilent Technologies Inc., Santa

Clara, CA). First-strand cDNA is synthesized using the SuperScript III First-Strand Synthesis SuperMix (Life Technologies, Rockville,MD). Quantititative qRT-PCR is performed using the ABI QuantStudio 7-Flex Detection System (Applied Biosystems, Foster City,CA). 18S rRNA, GAPDH and β-actin was used for normalization. The C_t method is used to calculate the relative expression by $2^{-\Delta\Delta Ct}$ using tibial muscle collected during the defect surgery.

For data set 1 (smooth spacer), PCR was conducted on sample of 125 ng of cDNA and for data set 2 (Textured spacer), PCR was conducted on sample of 1µg. After analyzing set 1, the amount of cDNA was considered to be possibly too low. Therefore, in set 2 (textured samples), the protocol was changed to load the PCR reaction with 1µg.

Statistical analysis was performed using generalized linear mixed model with Bonferroni correction using SAS 9.4 software (SAS Institute, Cary, NC). The response variable was each gene, the treatments were membrane position (Distal vs. Proximal) and membrane location (Inner vs. Outer). The effect of goat was included as a random factor. Statistical significance was determined for $p<0.0026$.

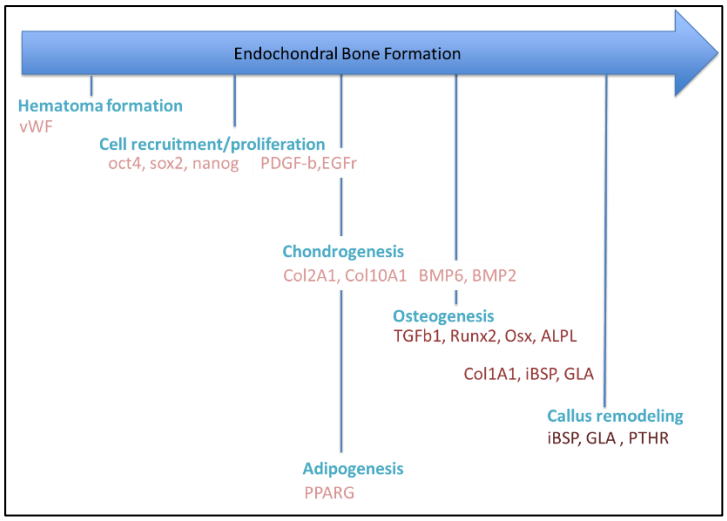


Figure 11 – Genes involved in the Endochondral Bone Repair Process

Gene	Function
VWF	Encodes a glycoprotein which functions as an antihemophilic factor carrier and a platelet-vessel wall mediator in the blood coagulation system. It is crucial to the hemostasis process.
OCT4	Encodes a transcription factor containing a POU homeodomain that plays a key role in embryonic development and stem cell pluripotency.
SOX2	Encodes a member of the SRY-related HMG-box (SOX) family of transcription factors involved in the regulation of embryonic development and in the determination of cell fate.
NANOG	Transcription factor involved in maintaining pluripotency in embryonic stem cells.
PDGFB	Protein encoded by this gene is a member of the platelet-derived growth factor family.
EGFR	Encodes a transmembrane glycoprotein that is a member of the protein kinase superfamily. This protein is a receptor for members of the epidermal growth factor family.
COL2A1	Encodes the alpha-1 chain of type II collagen, a fibrillar collagen found in cartilage.
COL10A1	Encodes the alpha chain of type X collagen, a short chain collagen expressed by hypertrophic chondrocytes during endochondral ossification.
BMP6	Part of a family of secreted signaling molecules that can induce ectopic bone growth. This gene has a proposed role in early bone development.
BMP2	Encodes a protein that acts as a disulfide-linked homodimer and induces bone and cartilage formation.
TGFB1	A polypeptide member of the transforming growth factor beta superfamily of cytokines.
RUNX2	Essential for osteoblastic differentiation and skeletal morphogenesis and acts as a scaffold for nucleic acids and regulatory factors involved in skeletal gene expression.
OSTERIX	Bone specific transcription factor that is required for osteoblast differentiation and bone formation.
ALPL	Encodes a membrane bound glycosylated enzyme, a proposed function of this form of the enzyme is matrix mineralization.
COL1A1	Encodes a fibril-forming collagen that is found in most connective tissues and is abundant in bone, cornea, dermis and tendon.
IBSP	Protein encoded by this gene is a major structural protein of the bone matrix
GLA	Encodes a homodimeric glycoprotein that hydrolyses the terminal alpha-galactosyl moieties from glycolipids and glycoproteins.
PTHR	A member of the G-protein coupled receptor family 2 that is a receptor for parathyroid hormone (PTH) and for parathyroid hormone-like hormone (PTHLH).
PPARG	Protein encoded is a regulator of adipocyte differentiation.

Table 1. Gene Analysis in Goat Induced Membranes.

Results:

Histology Results (Figure 12): All IM samples had a similar appearance in each treatment groups. Each IM sample included two relatively distinctive layers. Immediately adjacent to the spacer, a narrow inner zone was composed of dense mononuclear cells that appeared to be composed primarily of macrophages (foreign body response) with rare inflammatory cells. Variable numbers of small-caliber blood vessels were also present in the inner zone. The remainder of the tissue (outer section) was composed of fibrous connective tissue that extended to the deep margin of the section. All samples exhibited diffuse trichrome positivity indicative of the presence of fibrous connective tissue.

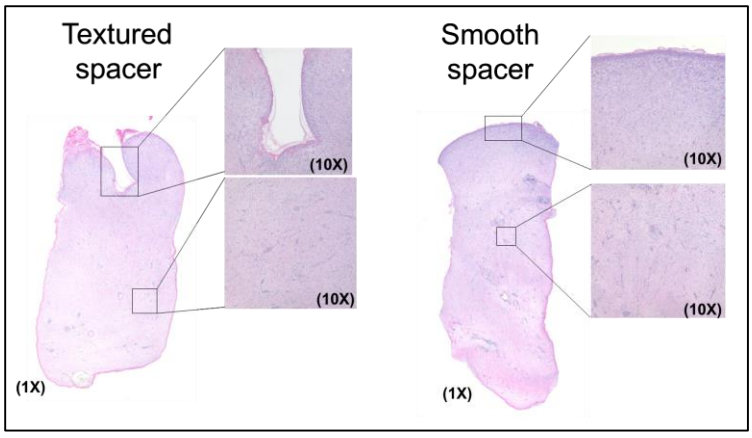


Figure 12 – Example of IM histology slide stained with H & E in textured spacer group and the smooth spacer group.

Cell/CTP analysis Results:

	Smooth Spacer		Textured Spacer	
	Inner IM N= 12	Outer IM N=12	Inner IM N= 16	Outer IM N=16
Cell Yield (Million cells per gram)	25.4 ± 5.5	11.2 ± 2.4	14.9 ±1.8	6.9 ±1.1
Total cell count after 6 Day culture (cells per cm^2)	4043.4 ± 1336.7	3662.0 ± 1272.1	5441.5 ± 1275.9	5272.1 ± 1015.2

Table 2 - Cellularity, total cells after 6 Day culture obtained in each group. Data are expressed as Mean ± Standard Deviation

Cellularity:

The number of nucleated cells per gram of digested membrane tissue varied significantly between smooth and texture groups as shown in Table 2. Mean inner membrane cellularity for smooth spacer group was 25.6 ± 18.7 million cells per gram while it was 14.9 ± 7.0 million cells per gram for the textured spacer group. Mean outer membrane cellularity for smooth spacer group was 11.2 ± 8.5 million cells per gram while it was 6.8 ± 4.5 million cells per gram for the textured spacer group.

The smooth spacer group had statistically significantly higher number cells per gram than textured membrane group for both inner and outer membranes.

Inner IM samples had significantly higher cell yield per gram than the outer IM by approximately two-fold in both smooth and textured spacer groups (table 2).

Colony:

Both inner and outer IM samples cultured under osteogenic differentiation media at Day 6 showed confluent colonies for most samples with large numbers of free cells and development of nodular structures, which appeared as highly dense, multi-layer clusters of cells. Therefore, no CTP Prevalence was obtained; only CTP progeny cell count at Day 6 was measured. These nodules were also found to be mineralized as demonstrated by positive alizarin red staining, a technique commonly used to detect calcium deposition in osteoblast cultures.

The number of CTP progeny cells at Day 6 varied widely between samples. Mean inner membrane CTP progeny cells at Day 6 for smooth spacer group was 4043.4 ± 1336.7 cells per cm^2 while it was 5441.5 ± 1015.2 cells per cm^2 for the textured spacer group. Respectively, mean outer membrane CTP progeny cells at Day 6 for smooth spacer group was 36262.0 ± 1275.9 cells per cm^2 and 5272.1 ± 1015.2 for the textured spacer group.

CTP progeny cell at Day 6 were comparable for both inner and outer samples. There was a trend for higher CTP progeny cell count at Day 6 in the textured spacer group than in the smooth spacer group but it was not significant.

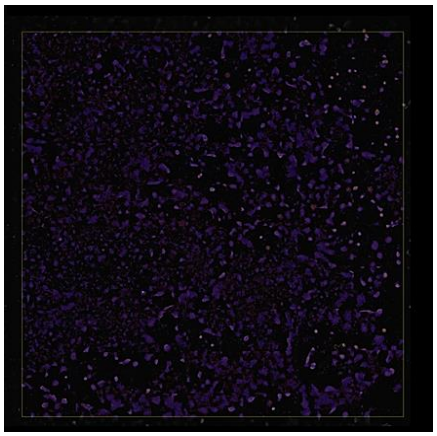


Figure 13 – Representative example of IM cell culture chambers at Day 6 showing many cells and confluent colonies.

For 4 goats, we were able to measure colony count at Day 6. The results are reported in figure 13 below. Based on the 4 goats where we were able to count colonies, we measured that mean CTP Prevalence was 4214.24 ± 4409.85 colonies per million cells plated and the number of cells per colony and colony size for goat membrane samples was very variable, 1 colony can contain between 13 up to 213 cells. Similar findings are also found in human iliac crest bone marrow aspirate sample.

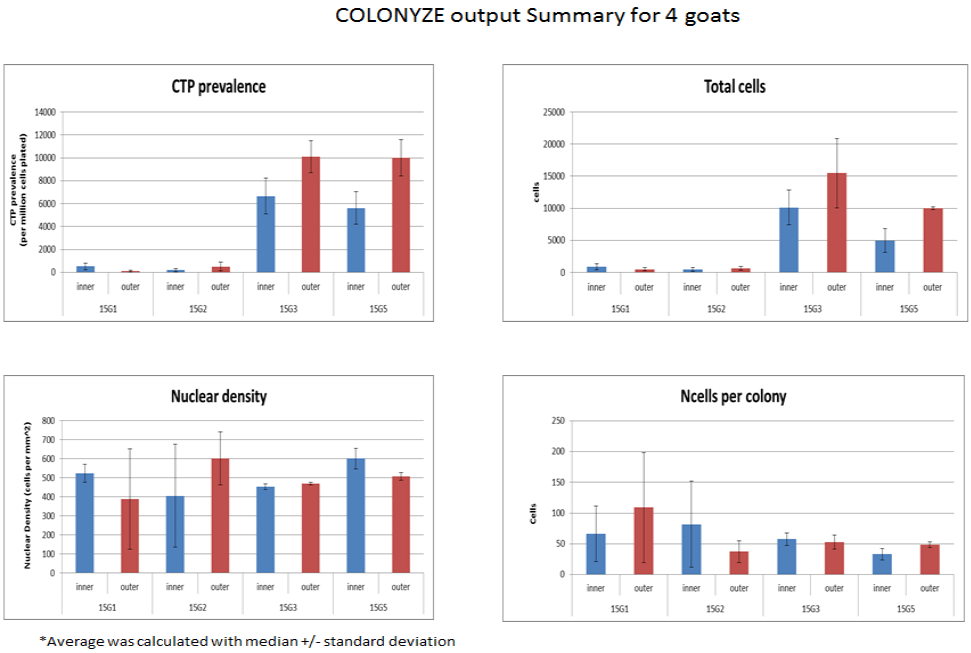


Figure 14 – CTP Prevalence, nuclear density, cells per colony, total cells at Day within chambers for the four goats where we were able to count colonies.

The induced membrane has biological characteristics and properties that may be used for improving the outcome of bone tissue engineering purposes. However, the cell plating density remains to be optimized for colony analysis. Further studies are still needed to optimize the osteogenic features of the induced membrane. We recommend a lower plating density of 10,000 cells per chamber (= 5,000 cells per ml) is now recommended for future studies so further characterization of the induced membrane can be made.

<u>Gene expression Results:</u>

Data set 1 (Figure 15) showed more variations which might reflect problems when processing these samples. The low amount of cDNA (125 ng) used to load per plate might have been too low resulting in non-detectable levels of some genes and therefore led to greater variations in the data. Data Set 1 are reported here but no conclusive analysis was drawn from this set. Conclusions were only made on the analysis of set 2 (textured spacer) (Figure 16).

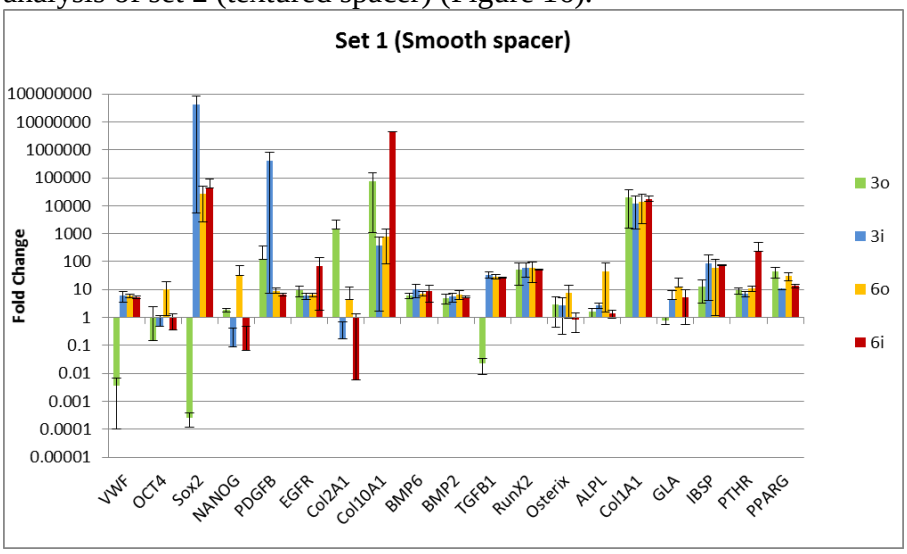


Figure 15 - Gene expression of membrane sections normalized to goat muscle after smooth spacer implantation (Set1). Each Bar represent Mean Fold change ± Standard Deviation.

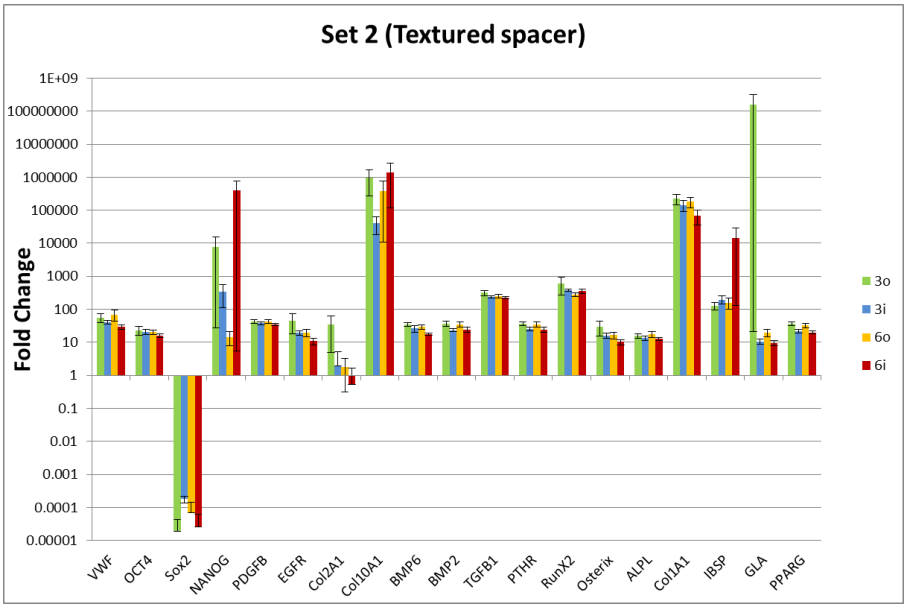


Figure 16 – Gene expression of membrane sections normalized to goat muscle after textured spacer implantation (Set2). Each Bar represent Mean Fold change ± Standard Deviation.

Comparison IM vs. Muscle (Table 3): The gene expression analysis from Data set 2 (texture surface) was loaded with 250 ng cDNA. These data demonstrated less variation than set 1. These data showed upregulation of target genes relative to normal muscle, with the one exception of Sox2 (gene involved in cell-fate). Sox2 was found to be decreased by approximately 210 fold.
13 of 19 genes of IM samples were significantly different than normal muscle. These included RunX2, Osterix, PPARG, OCT4, ALPL, EGFR, VWF, PDGFB, TGFB1, BMP2, BMP6, Col1A1, and PTHR. We noted that Col1A1 gene fold change was significantly high (95% Confidence Interval after Bonferroni correction was 76600 to 241500 fold). RunX 2 and TGFB1 genes had a fold change in the order 100 fold. All other genes had moderate increase (less than 35 fold).These results indicated evidence of genes within the IM cell population that has osteogenic and chondrogenic potentials.

	Set 2 = Textured						
	Mean Fold change (including all inner,outer, distal,proximal)	Median	SD	SE	SE after Bonferroni correction	Significance (adjusted after Bonferroni) Confidence interval lower limit	Confidence interval upper limit
VWF	50.9	35.1	60.4	7.8	21.8	29.2	72.7
OCT4	20.0	14.4	16.7	2.2	6.0	14.0	26.0
Sox2	-32371.3	-209.7	109064.8	14080.2	39283.7	-71655.1	6912.4
NANOG	93373.3	5.5	707543.3	91343.4	254848.1	-161474.8	348221.4
PDGFB	38.6	36.4	16.6	2.1	6.0	32.6	44.5
EGFR	25.1	12.6	56.8	7.3	20.5	4.6	45.5
Col2A1	8.4	-1.2	57.3	7.4	20.6	-12.2	29.0
Col10A1	1020688.3	62.1	3805525.9	491291.1	1370702.1	-350013.8	2391390.4
BMP6	27.0	23.0	18.0	2.3	6.5	20.5	33.5
BMP2	29.5	25.8	19.8	2.6	7.1	22.4	36.7
TGFB1	259.7	224.3	145.4	18.8	52.4	207.4	312.1
PTHR	29.7	24.5	19.5	2.5	7.0	22.6	36.7
RunX2	411.9	297.6	671.7	86.7	241.9	170.0	653.8
Osterix	18.4	13.7	28.8	3.7	10.4	8.0	28.8
ALPL	15.1	12.6	10.0	1.3	3.6	11.5	18.7
Col1A1	159024.6	23339.7	228951.1	29557.4	82465.3	76559.3	241489.9
IBSP	3488.1	91.1	25804.4	3331.3	9294.4	-5806.3	12782.5
GLA	39695877.8	9.9	307482829.5	39695845.6	110751409.3	-71055531.5	150447287.1
PPARG	26.9	22.8	16.5	2.1	5.9	21.0	32.9

Table 3 - Table summarizing overall statistical analysis of genes in the IM compared to muscle. Genes that showed significant difference with 2 < fold change < 35 were highlighted in yellow; with 150 < fold change < 500 in orange and fold change > 50,000 in purple.

Comparison between proximal and distal samples (Table 4): No significant main effects were found for target gene fold-change between distal or proximal sections in the data set 2.

	Set 2 = Textured					
	Distal Mean Fold change	SD Inner	Proximal Mean Fold change	SD Outer	Mean Difference (Distal - Proximal)	p value
VWF	48	48.7	53.9	71	-5.9	0.645
Oct4	21.7	21	18.3	11	3.4	0.3892
Sox2	-35005.6	131265.3	-29737.1	83399.5	-5268.5	0.8564
NANOG	4020.7	21012.8	182725.8	1000768.6	-178705.1	0.3311
PDGFB	39	16	38.1	17.3	0.9	0.8227
EGFR	32	78	18.1	19.5	13.9	0.3228
Col2A1	16.7	80.7	0.1	4.8	16.6	0.2037
Col10A1	682733.9	2144518.9	1358642.7	1020688.3	-675908.8	0.72228
BMP6	29.7	21.2	24.3	14.1	5.4	0.2297
BMP2	29.3	20.2	29.7	19.8	-0.4	0.9303
TGFB1	265.9	169.4	253.6	119.4	12.3	0.695
PTHR	29.9	17.8	29.4	21.4	0.5	0.9134
RunX2	491.1	934.9	332.7	175.5	158.4	0.3542
Osterix	22.8	38.6	14	12.7	8.8	0.2147
ALPL	14.3	9	15.9	11	-1.6	0.4611
Col1A1	182452.2	252952.2	135596.9	203775.4	46855.3	0.1334
IBSP	172	172.2	6804.2	36495.4	-6632.2	0.3257
GLA	79391740.6	434846388	15.1	14.7	79391725.5	0.3237
PPARG	28	16.3	25.9	16.9	2.1	0.5698

Table 4 - Table summarizing overall statistical analysis of IM gene Fold change comparing distal sections to proximal sections. No statistical difference between distal and proximal was detected in any genes.

Comparison between inner sections and outer sections (Table 5): The data analysis of set 2 showed that gene expression in the inner surface (1 mm) of the IM differs from the rest of the IM. Overall, the fold change in the inner samples was lower than in the outer sample.
No significant differences between inner and outer sections were detected after the Bonferroni adjustment except for PPARG. The fold change for PPARG (involved in adipocyte differentiation and decrease of anti-inflammatory response) in the inner sample was approximately 14.7 fold lower than the fold change in the outer sample (p-value=0.0011). Five genes including TGFB1, BMP2, BMP6, Col1A1 and PTHR showed trends that approached statistical difference, with values of fold change that were lower in the inner layer of the IM than in the outer layer. Each of these genes are considered to be pro-osteogenic. It is therefore possible to speculate that the removal of the inner-most layer of the IM

where these genes were expressed at lower levels may be a mechanism that contributes to the finding of increased bone formation in the scraped IM defect sites.

	Set 2 = Textured					
	Inner Mean Fold change	SD Inner	Outer Mean Fold change	SD Outer	Mean Difference (Inner - Outer)	p value
VWF	35.6	20.5	66.3	80.8	-30.7	0.0271
Oct4	18.8	11.4	21.2	20.9	-2.4	0.5464
Sox2	-25942.0	96710.2	-38800.7	121499.1	12858.7	0.6596
NANOG	182885.2	1000738.7	3861.3	21033.7	179023.9	0.3302
PDGFB	35.1	11.6	42.0	20.0	-6.9	0.0818
EGFR	14.9	11.4	35.3	78.9	-20.4	0.1563
Col2A1	-1.2	3.1	18.0	80.5	-19.2	0.2682
Col10A1	847749.0	3392663.0	1193627.7	4229822.0	-345878.7	0.4909
BMP6	21.3	16.4	32.8	18.0	-11.5	0.0189
BMP2	23.1	11.3	35.9	24.2	-12.8	0.0099
TGFB1	219.9	54.2	299.6	191.9	-79.7	0.0214
PTHR	23.3	14.1	36.0	22.2	-12.7	0.0076
RunX2	365.0	164.2	458.8	941.5	-93.8	0.5793
Osterix	13.7	9.4	23.2	39.4	-9.5	0.1808
ALPL	13.4	7.2	16.7	12.1	-3.3	0.1383
Col1A1	115731.9	175080.7	202317.2	268540.0	-86585.3	0.0107
IBSP	6841.5	36488.3	134.7	180.7	6706.8	0.3205
GLA	9.8	5.8	79391745.8	434846387.0	-79391736.0	0.3237
PPARG	19.6	8.3	34.3	19.3	-14.7	0.0011

Table 5 - Table summarizing overall statistical analysis of IM gene Fold change comparing inner sections to outer sections. Genes that showed significant difference with p>0.0026 were highlighted in green; trend for difference with 0.0026<p<0.05 were highlighted in yellow.

c) Predictive modeling (NMRC)

Protocol:

Data preprocessing -

a) ***Missing data*:** Most modeling algorithms fail when the data contains missing values except for a few (e.g. Bayesian based algorithms). The data collected contains around 1% of missing data which were imputed using missforest in R. 'missForest' is used to impute missing values particularly in the case of mixed-type data. It can be used to impute continuous and/or categorical data including complex interactions and nonlinear relations. It yields an out-of-bag (OOB) imputation error estimate [9].

b) ***Formatting the variables:*** In order to run logistic regression, on the scraping variable and spacer texture, they were converted to factor with 2 levels: 0 and 1. The data were partitioned between training and test set in a proportion of 70% training and 30%. The small sample size of 32 compelled us to test the model with a 50:50 proportion.

c) ***Modeling the effects of spacer texture on collected:*** Setting spacer texture as the response variable, a naïve Bayes based model was built on the training data. The standard naive Bayes classifier relies on the assumption of independence of the predictor variables, and of metric predictors that are normally distributed (Gaussian distribution). Sensitivity and specificity test were performed on the model performance to discriminate among spacer texture. Individual experiment was model to see their effects on spacer texture. A logistic regression was done and for numerical data, and variables with a significant p value were retained for model building.

1. **Effects of spacer texture on CT-2.5 (ie total bone volume in 2.5 cm central region) and X-Ray ranking values:** A logistic regression modeling was performed to look at the effects of spacer texture and membrane scraping on CT-2.5, X-ray.
2. **Effects of spacer texture in predicting IM Biology**: The effects of nine biological metrics to predict spacer texture were analyzed using a logistics regression technique (Table IM

Biology**)**. Variables with a p-value < 0.05 was considered significant in predicting spacer texture.

3. **Effects of the presence of spacer texture on gene expressions in the inner section**: The expression of nineteen genes located in the inner IM section were investigated in relation with the presence of spacer texture using logistic regression. Important genes expressed due to spacer texture were selected in model building (gene expression table). Due to the quality of the data which appeared unscaled, and impossible to be analyzed by logistic regression, we resolved to select my other methods the top variables important in building model around the spacer texture. Random forest based model was built to select important genes for predictive modeling. Random forest is primarily an ensemble learning method for classification, and regression (Brenan. Others model building techniques such as **Boruta** [10], **Lasso** [11], **pda** PDA (penalized discriminant analysis), **nnet** (Neural network) were run to select important genes. Importance variables returned will be used to build final model. Boruta is an R package which acts as a wrapper algorithm for relevant feature selection. It finds important variable by comparing original attributes importance with the importance achievable at random, estimated using their permuted copies and progressively eliminating irrelevant features to stabilize the test. By default Boruta uses random forest for computing. Lasso is a shrinkage and selection method for linear regression. It minimizes the usual sim of squared errors with a bound on the sm of the absolute values of the coefficients. It is considered more stringent than logistic regression. PDA (penalized discriminant analysis) is a data analytic tool for studying the relationship between a set of predictors and a categorical response. It is used in cases where there are many highly corrected predictors. It is also used for dimension reduction. Nnet (Neural network), used in the R package, is a feed-forward neural networks with a single hidden layer.
4. **Effects of the presence of spacer texture on gene expressions in the outer section:** The same analysis pathway used to select important genes expressed will be applied to genes expressed in the outer section.

d) ***Modeling the effects of membrane scraping on outcome variables:*** The same approach to feature selection which was applied to spacer texture was used in modeling for membrane scraping.
e) ***Predictive Modeling of the Effects of IM biology on microCT-2.5 level:*** Direct linear effect of Scraping on the level of CT-2.5 was investigated. A linear regression equation was designed to answer to question of any potential relation between scraping and CT-2.5 level.
f) ***Effects of gene expression in the inner and outer section on the microCT-2.5 level:*** A linear regression model was designed to select the genes that were significantly expressed in relation to CT 2.5 level.
g) ***Effects of Graft, cellularity on microCt-2.5 level:*** Linear regression equations were generated to model the contributions of ACBG variables, IM cellularity variables on CT-2.5 level. Features which contributed significantly to CT-2.5 level were retained in the final predictive model.

Data Modeling Protocol:

A table containing a total of 32 observations of data collected from various experiments were pooled in a spreadsheet. It contained data on goat (age, weight), data on the surgeons who performed the surgeries, the quantitative histology of the IM tissues graft, the quantitative histology of the tibia defect, the expression profiles of 19 genes in the inner and outer membrane sections, the CT scan of the tibia defect, the Xray ranking. The data were organized to match each goat for easy analysis. The animal ages range from 3 to 5 years, and weight 41-65 kg. Four different surgeons dissected, scraped the membrane and implanted the spacers in the animals. Goats that were grouped as membrane scraped or not (0, for not scraped and 1 for scraped), and spacer texture. Sixty nine features were collected from the 32 goat in the study for a total of 2208 data entries in the database. Of the 2208 data collected, 8 were missing which were subsequently imputed using missForest algorithm. The table contained mixed data types: integers, numerical, and factor.

a) **Predicting microCT total bone value from the predictors collected:** A regression analysis using CT-2.5 values as the response variable from the experiments performed on the goats was done. Due to the various data structure, we grouped the data by experimental type:
    a. Group 1 included scraping parameter, spacer texture parameter, microCT-5 values (ie total bone volume in 5 cm defect region), and X-Ray rank.
    b. Group 2 comprised of the IM histology features (Tib Histo Bone Area, IM Histo Vascularity, IM Histo Fibrosis, IM Histo inflammatory Cellularity, IM Histo Thickness, IM membrane weight, IM Inner.Mean.cellularity, and Expanded IM inner.cells).
    c. Group 3 was the genes located in the inner section of the membrane tested.
    d. Group 4 comprised genes located on the outer section of the membrane
    e. Group 5 comprised ACBG and BMA biology parameters. The variable included in the model building were BMA CTP Prevalence, MS cellularity, MS CTP Prevalence, TS cellularity, TS CTP Prevalence , MS CTP Prevalence, Graft Histo cartilage, Graft Histo bone, Graft Histo hematopoetic marrow, Graft Histo adipose tissue, and Graft Histo vasculature area.

b) Effects of spacer texture and membrane scraping on CT-2.5 level and X-ray rank: We wanted to know if spacer texture and membrane scraping could predict CT-2.5 level and X-ray ranking. The response variable membrane scraping and spacer texture both binary outputs so we ran logistic regression models.

c) Effect of spacer texture on IM biology.

Results:

1. Data compiled from the IM goat study

The code ran in 0.24 seconds. After the missing data were successfully computed, modeling could be performed. The total observations were partitioned 70% for training data and 30% for testing the model. The partition were achieved using the function create partition from the caret packages in R which splits the data 24 observations for building the model and 8 for testing its predictive performance.

2. Building a classification model around the spacer texture and membrane scraping.

Can all the features measure help sort out goat with spacer texture or not? Seventeen (17) goat had spacer texture and 15 did not. In order to answer this question, and naïve Bayes function was run on the whole data.

The model was sensitive at 93%, 100% specific, and PPV was 100% and NPV was 94.4%. Of the 8 test data, 4 was group as 0, and 3 as 1.

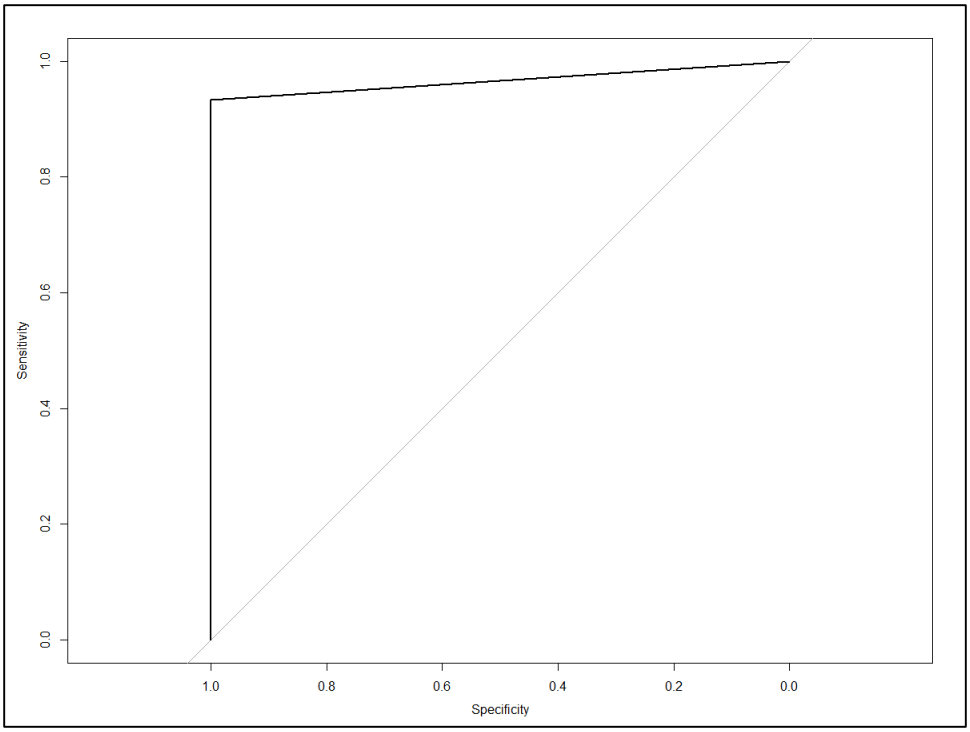


Figure 17 - ROC curve for Spacer Texture model:The model using all observed predictors was 93% sensitive and 94% specific.

Of the 32 observations, 16 had their membrane scraped and 16 their membrane not scraped. Using all the features collected, we built a predictive tool to predict goats with membrane scraped or not. The model performed at 100% sensitivity and 54% specificity for PPV of 70% and NPV of 100%.

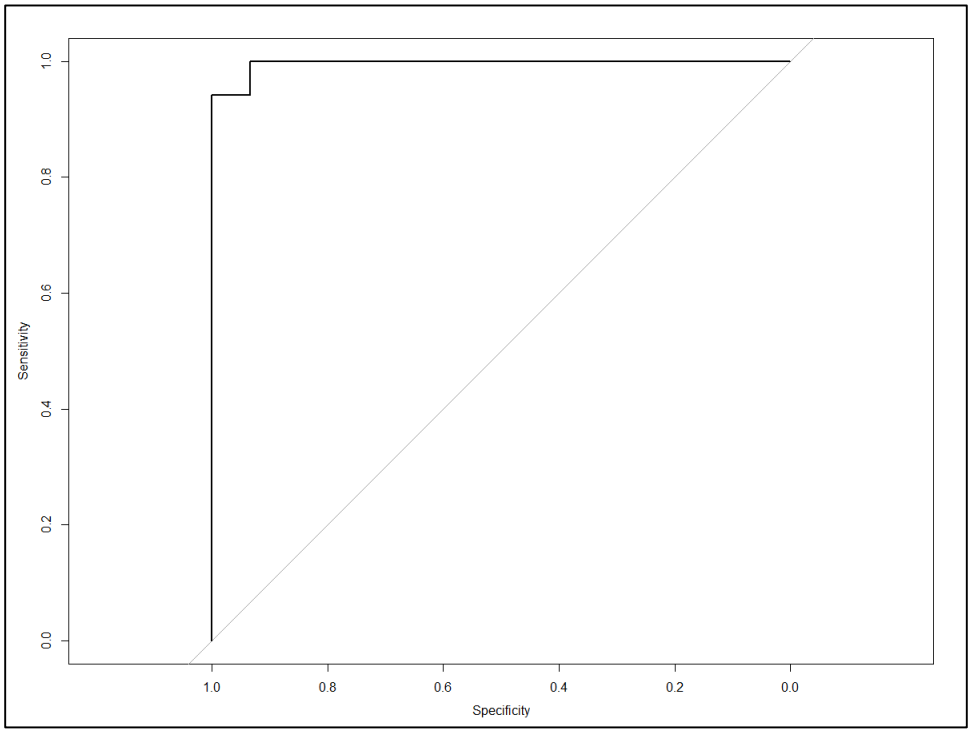


Figure 18 – ROC curve for membrane scraping model. The model performed at 100% sensitivity and 54% specificity for PPV of 70% and NPV of 100%.

3. **Predicting CT scan value from the predictors collected**

A regression analysis using CT-2.5 (total bone volume in 2.5 cm central region) values as the response variable from the experiments performed on the goats was done. Due to the various data structure, we grouped the data by experimental type.

Group 1 included scraping, spacer texture, CT-2.5. CT-5, and X-Ray rank. The model looked at these variables to predict CT-2.5. We found that CT.5 (total bone volume in 5 cm defect region) was a better predictor of CT-2.5 values at the p value = 1.9e-15.

Estimate	**Std.**	**Error**	**t-value**	**Pr(>|t|)**
(Intercept)	-192.636	255.3244	-0.754	0.457
scraping1	20.13033	70.2536	0.287	0.777
Spacer.texture1	54.75502	69.36942	0.789	0.437
CT.5	0.49048	0.03024	16.222	**1.90E-15**
Xray.Rank	-5.33292	9.86053	-0.541	0.593

Table 6 - Linear regression metrics for CT-2.5 model for group 1.

Group 2 comprised of the IM histology features (Tibia Histology Bone Area, IM Histology Vascularity, IM Histology Fibrosis, IM Histology inflammatory Cellularity, IM Histology Thickness, IM membrane weight, IM Inner Mean.cellularity, and Day 6 IM inner and outer cell count). We found Tibia histology bone area to be significant predictor (p= 0.00762) of CT-2.5 level than the others features tested in this group 2.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-1.195e+03	6.455e+03	-0.185	0.85491
Tib_Histo_Bone.Area..3.5	2.927e+00	9.918e-01	2.952	**0.00762**
IM__Histo_Vascularity	-4.847e+02	5.613e+02	-0.863	0.39763
IM_Histo_Fibrosis	1.930e+02	1.640e+03	0.118	0.90746
IM_Histo_inflammatory_Cellularity	-8.182e+02	6.398e+02	-1.279	0.21493
IM_Histo_Thickness	1.578e+02	1.238e+02	1.275	0.21627
IM.membrane._.weight	2.750e+03	3.529e+03	0.779	0.44450
IM_Inner.Mean.cellularity	1.916e+01	1.772e+01	1.081	0.29188
Day 6.IM_inner.cells	7.688e-03	6.773e-02	0.114	0.91070
IM_Outer.Mean.cellularity	-5.130e-01	4.730e+01	-0.011	0.99145
Day 6.IM_outer.cells	2.911e-02	8.575e-02	0.339	0.73763

Table 7 - Linear regression metrics for CT-2.5 model for group 2.

Group 3 was the genes located in the inner section of the induced membrane (IM). A regression analysis was performed to find which gene is significantly predicted CT-2.5 level. Of the nineteen genes located on the inner IM section, none was a significant predictor for CT-2.5 level.

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.205e+03  1.398e+03   0.862    0.406
## IM_inner_RunX2  3.372e+00  9.111e+00   0.370    0.718
## inner_Osterix   1.234e+01  1.539e+02   0.080    0.937
## inner_PPARG    -3.123e+01  6.238e+01  -0.501    0.626
## inner_OCT4      1.001e+02  8.139e+01   1.230    0.242
## inner_Sox2      3.395e-06  4.176e-06   0.813    0.432
## inner_NANOG    -3.883e-05  8.662e-04  -0.045    0.965
```

```
## inner_ALPL      3.089e+01  9.808e+01   0.315    0.758
## inner_EGFR      2.828e+00  2.711e+00   1.043    0.317
## inner_VWF      -9.562e+01  6.040e+01  -1.583    0.139
## inner_PDGFB    -2.011e-03  4.525e-03  -0.444    0.665
## inner_TGFB1    -1.727e+00  8.247e+00  -0.209    0.838
## inner_BMP2      8.455e+01  5.804e+01   1.457    0.171
## inner_BMP6     -5.809e+01  5.965e+01  -0.974    0.349
## inner_Col1A1    5.292e-03  7.370e-03   0.718    0.486
## inner_Col2A1   -9.295e+01  1.080e+02  -0.861    0.406
## inner_Col10A1   1.693e-04  3.938e-04   0.430    0.675
## inner_GLA      -6.706e+01  1.179e+02  -0.569    0.580
## inner_IBSP     -2.056e-02  2.162e-02  -0.951    0.361
## inner_PTHR      6.494e+00  5.859e+00   1.108    0.289
```

Table 8 - Linear regression metrics for CT-2.5 model for group 3.

Group 4 comprised genes located on the outer section of the IM. A regression model to predict the expression of the genes in response of CT-2.5 level revealed that five genes were impacted by CT-2.5. Increased expression of Col1A1 (p=0.0354) and BMP2 (p=0.0264) had a positive effect on CT2.5 with a p-value less than 0.05 while Sox2 (p=0.0860), BMP6 (p=0.0557), Col10A1 (p=0.0894) and GLA (p=0.0764) were near significant. However, increased expression of BMP6 and GLA were associated with decreased CT2.5.

```
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.218e+03  6.995e+02   1.741   0.1072
## outer_RunX2    1.646e+00  6.573e+00   0.250   0.8065
## outer_Osterix  4.684e+01  1.013e+02   0.462   0.6521
## outer_PPARG    1.369e+01  1.564e+01   0.876   0.3985
## outer_OCT4    -8.033e+00  6.071e+01  -0.132   0.8969
## outer_Sox2     9.735e-03  5.205e-03   1.870   0.0860 .
## outer_NANOG    1.826e-02  3.739e-02   0.488   0.6342
## outer_ALPL    -1.546e+01  1.476e+01  -1.048   0.3155
## outer_EGFR    -4.765e+01  6.237e+01  -0.764   0.4596
## outer_VWF      2.579e+00  2.173e+00   1.187   0.2583
## outer_PDGFB   -4.273e-01  8.039e-01  -0.532   0.6048
## outer_TGFB1   -8.250e+00  8.098e+00  -1.019   0.3284
## outer_BMP2     1.070e+02  4.228e+01   2.530   0.0264 *
## outer_BMP6    -9.506e+01  4.487e+01  -2.119   0.0557 .
## outer_Col1A1   6.920e-03  2.919e-03   2.371   0.0354 *
## outer_Col2A1   1.138e-01  1.150e-01   0.990   0.3417
## outer_Col10A1  4.974e-04  2.692e-04   1.848   0.0894 .
## outer_GLA     -4.847e-06  2.500e-06  -1.939   0.0764 .
## outer_IBSP     2.194e+00  6.783e+00   0.323   0.7519
## outer_PTHR     2.232e+00  4.402e+01   0.051   0.9604
```

Table 9 - Linear regression metrics for CT-2.5 model for group 4.

Group 5 investigated the effects of graft biology on CT-2.5 response. The variable included in the model building were Bone Marrow Aspirate (BMA) cellularity and CTP Prevalence, Marrow Space (MS) cellularity and CTP Prevalence, Trabecular Surface (TS) cellularity and CTP Prevalence , ACBG

Histology cartilage area, Graft Histology bone area, Graft Histology hematopoietic marrow area, Graft Histology adipose tissue area, and Graft Histology vasculature area.
The features that were significant predictors of CT-2.5 level are CTP Prevalence for Trabecular surface of ACBG graft with a p-value= 0.00424, and near significant features are CTP Prevalence of Bone Marrow aspirate (p=0.07547), and ACBG Histology adipose tissue amount (p=0.08598). Note that the effect of ACBG adipose tissue was negative, therefore increasing adipose histology was associated with a decrease in CT2.5. In contrast, increases in CTP prevalence among TS cells and in the BMA was associated with an increase in CT2.5.

BMA.cellularity	-7.7384	7.0756	-1.094	0.28709	
BMA.CTP.Prevalence	22.0851	11.7784	1.875	**0.07547**	.
MS.cellularity	5.8056	5.1634	1.124	0.27416	
MS.CTP.Prevalence	-11.0632	7.2596	-1.524	0.14318	
TS.cellularity	4.4504	6.6696	0.667	0.51223	
TS.CTP.Prevalence	2.7982	0.8675	3.225	**0.00424**	**
Graft_Histo_cartilage	-0.6553	21.4711	-0.031	0.97596	
Graft_Histo_.bone	45.5486	46.6194	0.977	0.34022	
Graft_Histo_hematopoetic.marrow	-14.7145	25.1369	-0.585	0.56484	
Graft_Histo_adipose.tissue	-3407.3018	1886.5932	-1.806	**0.08598**	.
Graft_Histo_vasculature.area	-254.3541	3743.0822	-0.068	0.94650	

Table 10 - Linear regression metrics for CT-2.5 model for group 5.

4. Effects of spacer texture and membrane scraping on CT-2.5 level and X-ray rank

We modeled if spacer texture and membrane scraping could predict CT-2.5 level and X-ray ranking. The response variable membrane scraping and spacer texture both binary outputs so we ran logistic regression models.
Membrane scraping did not significantly affect CT2.5- levels and X-ray Ranking as shown in table 11. No significant p value was found. Spacer texture did not significantly affect CT-2.5 and X-ray rank or vice versa as shown in table 12.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.9585444	2.7714776	-0.346	0.729
CT.2.5	0.0009173	0.0023906	0.384	0.701
CT.5	-0.0001429	0.0011803	-0.121	0.904
Xray.Rank	0.0198139	0.1119994	0.177	0.860

Table 11 - Logistic regression statistics for membrane scraping

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	4.281593	2.941563	1.456	0.146
CT.2.5	0.002070	0.002326	0.890	0.374
CT.5	-0.001507	0.001269	-1.188	0.235
Xray.Rank	-0.154293	0.113735	-1.357	0.175

Table 12 - Logistic regression statistics for spacer texture

5. Effect of spacer texture on IM biology.

We modeled the response of spacer texture to IM Biology as shown in table 13 using a logistic regression. We found that IM vascularity was near to significantly affected by spacer texture.

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.415e+01	1.582e+04	0.002	0.9983
Tib_Histo_Bone.Area..3.5	3.854e-03	4.161e-03	0.926	0.3543
IM__Histo_Vascularity	5.932e+00	3.110e+00	1.907	**0.0565**
.				
IM_Histo_Fibrosis	-1.020e+01	3.956e+03	-0.003	0.9979
IM_Histo_inflammatory_Cellularity	-8.033e-02	3.932e+00	-0.020	0.9837
IM_Histo_Thickness	-1.631e+00	1.021e+00	-1.598	0.1101
IM.membrane._.weight	2.428e+01	1.991e+01	1.220	0.2226
IM_Inner.Mean.cellularity	-5.237e-02	7.743e-02	-0.676	0.4988
Day6.IM_inner.cells	2.545e-04	2.339e-04	1.088	0.2766
IM_Outer.Mean.cellularity	7.689e-02	3.044e-01	0.253	0.8006
Day6.IM_outer.cells	3.979e-04	5.917e-04	0.673	0.5013

Table 13 - Logistic regression metrics for spacer texture and IM biology predictors.

6. Effect of spacer texture on gene expression

Our modeling of genes expression due in relation to spacer texture using the logistics regression equation did not succeed. The scale of the data collected render the algorithm unapplicable. We resorted to applying various analytics techniques to overcome the poor data quality. Important genes selection was performed by running random forest, **Boruta**, **Lasso**, **pda**, and **nnet.**

Figure 19 shows in green the most important genes that were influenced by the spacer texture in the inner section.

The more stringent variable selection algorithm is shown to be Lasso (Glmnet) which selected only TGFB1, ALPL, PDGFB, OCT4, and IBSP as important genes expressed in the inner IM section (Table 14) affected by the spacer texture. The genes consistently ranked on the top of the list was used to build and predictive models of gene expression in the inner section. Table 14 shows in descending order the list of importance genes as sorted by the Random forest analysis and Table 15 shows in descending order the list of importance genes as sorted by neural network.

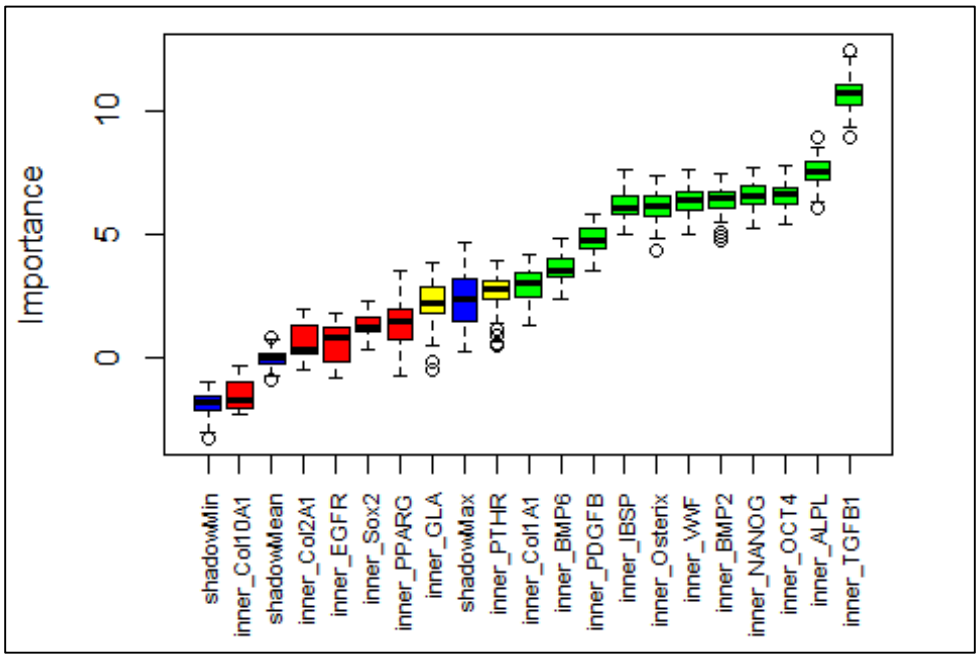


Figure 19 – Importance of genes in the inner section in response to spacer texture.

```
glmnet variable importance
##
##                Overall
## inner_TGFB1   100.000
## inner_ALPL     27.017
## inner_PDGFB    19.374
## inner_OCT4     14.761
## inner_IBSP      2.777
## inner_PTHR      0.000
## inner_VWF       0.000
## inner_Sox2      0.000
## inner_BMP2      0.000
## inner_PPARG     0.000
## inner_Col2A1    0.000
## inner_NANOG     0.000
## inner_Osterix   0.000
## inner_Col10A1   0.000
## inner_BMP6      0.000
## inner_EGFR      0.000
## inner_GLA       0.000
## inner_Col1A1    0.000
```

Table 14 - Important genes found ranked by Lasso (Glmnet)

```
## rf variable importance
##
##                Overall
## inner_TGFB1   100.000
## inner_OCT4     67.666
## inner_PDGFB    66.852
## inner_ALPL     63.674
## inner_IBSP     61.751
## inner_NANOG    61.396
## inner_VWF      60.564
## inner_Osterix  59.188
## inner_BMP2     53.059
## inner_Col1A1   30.023
## inner_BMP6     29.087
## inner_PTHR     24.297
## inner_GLA      21.041
## inner_Col2A1   14.807
## inner_Col10A1   9.303
## inner_EGFR      8.370
## inner_PPARG     3.780
## inner_Sox2      0.000
```

Table 15 - Important genes found ranked by Random Forest analysis

```
nnet variable importance
##
##                Overall
```

```
## inner_TGFB1   100.000
## inner_ALPL     70.821
## inner_OCT4     61.220
## inner_PDGFB    50.580
## inner_VWF      48.515
## inner_Col10A1  47.822
## inner_BMP2     37.931
## inner_Osterix  29.759
## inner_Col2A1   20.475
## inner_PPARG    19.714
## inner_EGFR     18.727
## inner_BMP6     17.232
## inner_PTHR     16.065
## inner_IBSP     15.546
## inner_NANOG    11.725
## inner_Sox2     10.468
## inner_Col1A1    6.657
## inner_GLA       0.000
```

Table 16 - Important genes found ranked by neural network

The same analytic approach to model design was applied to genes in the outer IM sections.
Boruta displays in Figure 20 the importance genes expression in the outer section. Table 17, 18, and 19 shows the lists of important genes that were significant predictors of spacer texture. The more stringent variable selection algorithm is shown to be Lasso (Glmnet) which selected only BMP6, TGFB1, BMP2, ALPL, PTHR, Col1A1, Sox2 as important genes expressed in the outer IM section affected by the spacer texture (Table 17).

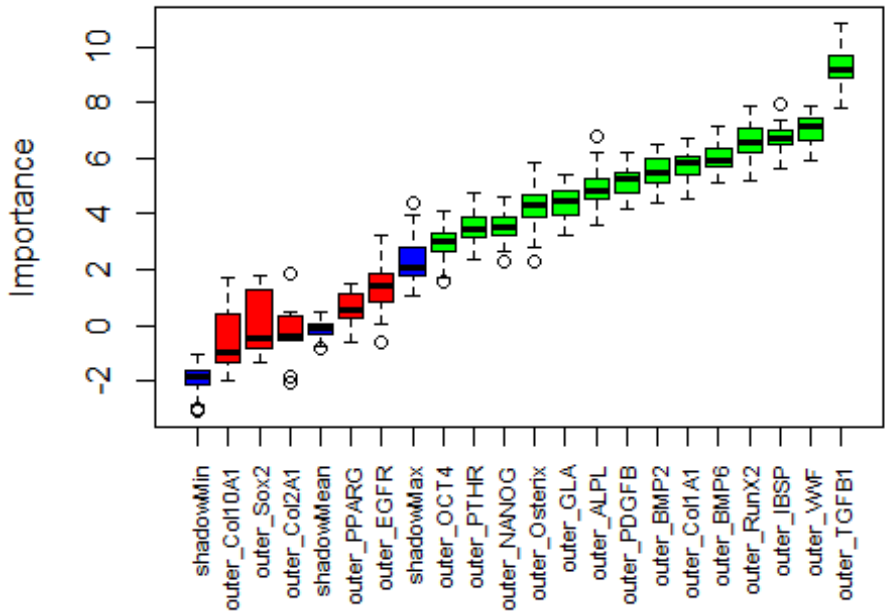


Figure 20 – Importance of genes in the outer section in response to spacer texture.

```
## glmnet variable importance
##
##                Overall
## outer_BMP6     100.00
## outer_TGFB1     65.76
```

```
## outer_BMP2      54.40
## outer_ALPL      28.92
## outer_PTHR      25.93
## outer_Col1A1    20.30
## outer_Sox2      14.43
## outer_Col10A1    0.00
## outer_PDGFB      0.00
## outer_GLA        0.00
## outer_VWF        0.00
## outer_OCT4       0.00
## outer_Osterix    0.00
## outer_IBSP       0.00
## outer_EGFR       0.00
## outer_PPARG      0.00
## outer_NANOG      0.00
## outer_RunX2      0.00
## outer_Col2A1     0.00
```

Table 17 - Important genes found ranked by Glmnet (Lasso)

```
##               Overall
## outer_TGFB1   100.000
## outer_IBSP     82.716
## outer_VWF      80.320
## outer_RunX2    73.534
## outer_GLA      65.553
## outer_BMP2     62.537
## outer_ALPL     61.800
## outer_BMP6     54.579
## outer_PDGFB    52.261
## outer_Osterix  51.092
## outer_Col1A1   48.436
## outer_PTHR     45.921
## outer_OCT4     35.961
## outer_NANOG    33.978
## outer_EGFR     17.501
## outer_Col2A1    7.987
## outer_PPARG     5.826
## outer_Sox2      4.712
## outer_Col10A1   0.000
```

Table 18 - Important genes found ranked by Random Forest

```
## nnet variable importance
##
##               Overall
## outer_TGFB1   100.000
## outer_OCT4     68.553
## outer_PPARG    67.519
## outer_BMP6     62.924
## outer_BMP2     56.754
```

```
## outer_RunX2    49.284
## outer_IBSP     39.539
## outer_Osterix  36.456
## outer_Sox2     33.667
## outer_PTHR     24.830
## outer_Col2A1   20.113
## outer_PDGFB    14.433
## outer_VWF      14.328
## outer_ALPL     13.649
## outer_Col1A1   11.827
## outer_EGFR      9.839
## outer_NANOG     7.592
## outer_GLA       5.417
## outer_Col10A1   0.000
```

Table 19 - Important genes found ranked by neural network

2.3 Nested development award (NMRC/WRNMC - Dr. D'Alleyrand)

Dr. D'Alleyrand had continued his interaction with Drs. Pluhar and Muschler and the veterinary staff at Cleveland Clinic during the animal surgeries in Aim 2. He became engaged and knowledgeable regarding the animal care process and participated in group laboratory discussions. He had received more specialized training and orientation to microCT analysis, both on a theoretical and practical level. He worked with raw CT images and gain operating experience with the process of moving from raw images to quantitative data, including sources of variation and error. He had participated intimately in the process of preparing summary data, presentation materials and contributed as an author and editor for manuscript submissions and give presentations in selective settings (e.g. OTA, SOMOS).

4. KEY RESEARCH ACCOMPLISHMENTS:

- All surgeries and all animal care in this project have been completed.
- All microCT data have been performed for Aim 1 and Aim2 and showed significantly increased bone formation in the scraped membrane group while there was no significant improvement of bone healing when textured spacer was used.
- Histomorphometry data from the Aim1 and Aim 2 tibia samples are consistent with the data obtained in the microCT analysis.
- The characterization of the cellularity and prevalence of osteogenic connective tissue progenitors from the induced membrane showed that individual induced membranes vary widely with respect to weight, cellularity and CTP prevalence. Inner membrane was more cellular than outer membrane but there was no difference in Day 6 cultured cell count.
- We demonstrated that the IM is composed of two distinctive portions: a thin glistening inner surface, comprised of richly cellular layer with some vascular vessels and an outer part made of fibroblasts and collagen. Of particular note, we demonstrated that the inner surface of the IM has higher levels of gene expression for PPARG and trends to lower expression of five other osteotropic genes (TGFB1, BMP2, BMP6, Col1A1 and PTHR). This may help explain why surgically removing this layer benefits bone regeneration.
- We demonstrated the value of predictive modeling in the CCTD model and the power of this model for dissection of the complex relationships between biological variables and outcome:
  - We demonstrated that IM membrane biology was associated with differences in outcome. In the outer section of IM, increased expression of Col1A1 (p=0.0354) and BMP2 (p=0.0264) had a positive effect on CT2.5 with a p-value less than 0.05 while Sox2 (p=0.0860), BMP6

(p=0.0557), Col10A1 (p=0.0894) and GLA (p=0.0764) were near significance. However, in increased expression of BMP6 and GLA were associated with decreased CT2.5.

- We demonstrated that ACBG and BMA quality were associated with significant differences in bone formation (CT2.5). Increases in CTP prevalence among TS cells and in the BMA was associated with an increase in CT2.5. Increases in ACBG adipose tissue composition was associated with a decrease in CT2.5.
- While spacer texture was not associated with a change in bone formation, it was associated with changes in gene expression within the IM, specifically TGFB1, ALPL, PDGFB, OCT4, and IBSP

5. CONCLUSIONS:

Experience to date clearly shows that the CCTD model "raises the biological bar" for bone regeneration research and development. These data establish the CCTD model as the most rigorous and clinically relevant large animal model for testing new bone regeneration materials and therapies.

This study highlights that the Chronic Caprine Tibial Defect Model proved to be sensitive to detect variations in:

- Surgical technique (scraping)
- Spacer Design (biomaterials, texture, drug delivery)
- Graft source quality (ACBG fat content and CTP prevalence and BME CTP prevalence)

Surgical removal of the inner surface of the IM significantly increased bone regeneration (p=0.041). This finding can be immediately translated into clinical practice.

Modifying the texture on the PMMA spacer to double the surface area of the IM did not result in improved bone regeneration. However, surface modification did change the gene expression within the resulting IM, leaving open the opportunity to examine other spacer approaches.

The CCTD model is particularly well suited to advance the field of clinical bone regeneration using cellular therapies, and optimization of clinically relevant methods for harvest and processing of CTP sources to enhance the concentration and prevalence CTPs in the site of bone regeneration. As these data demonstrated, ACBG and BMA harvested from the caprine sternum is comparable to human clinical tissues in terms of content and the concentration and prevalence of CTP. Moreover, the CCTD model is sensitive to changes in bone graft quality and the concentration and prevalence of CTPs.

The CCTD model is also well positioned to advance our understanding of the biological attributes of the IM and to explore means of optimizing these attributes through the use of spacers and other methods. Histological analysis of the induced membrane samples using a smooth PMMA spacer generates a relatively bland collagenous matrix of variable thickness with low cellularity and moderate vascularity, covered by a thin inner surface layer (foreign body response) with rare inflammatory cells, covering an underlying vascular plexus that varies in density. We also demonstrated that the IM is composed of two distinctive portions: a thin glistening inner surface, comprised of richly cellular layer with some vessels and an outer part made of fibroblasts and collagen. Of particular note, we demonstrated that the inner surface of the IM has higher levels of gene expression for PPARG and tends to have a lower expression of five other osteotropic genes (TGFB1, BMP2, BMP6, Col1A1 and PTHR). This may help explain why surgically removing this layer benefits bone regeneration.

Our understanding of the cellular biology of the IM remains an incomplete story. Cell analysis of the induced membrane demonstrated that it contains substantial number of cells, including CTPs with osteogenic and chondrogenic potential. We were able to measure CTP prevalence in only four animals, however, due to the fact that CTP prevalence in the IM was greater than expected and plated samples were overrun with cells making colony formation impossible to discern. The fact that the IM may be a

rich source of CTPs may be a good or a bad thing for bone regeneration, depending on the biological potential and tendency for commitment of the cells that are present. Further assessment into the nature of the cell population that is resident in the IM and their independent contribution to bone regeneration opens opportunities for targeting and optimization of this cell source for bone regeneration using the CCTD model.

6. PUBLICATIONS, ABSTRACTS, AND PRESENTATIONS:

a. Publication:

1. This research was featured as a highlighted study in the Congressionally Directed Medical Research Programs FY16 Annual Report.
2. Luangphakdy V, Pluhar GE, Piuzzi NS, D'Alleyrand JC, Carlson C, Bechtold J, Forsberg J, Muschler GF. The Masquelet induced membrane technique: Optimizing surgical technique and spacer design. Submitted on Nov. 30th, 2016 to Clinical Orthopaedics and Related Research.

b. Presentations:

1. Weinzierl A, Toth F, Pluhar GE, Muschler GF,Bechtold J, Luangphakdy V, Carlson C, Histomorphometric Comparison of Graft Efficacy Using a Caprine Chronic Tibial Defect Model. ORS, 2015, Las Vegas, Nevada, March 28-31, 2015.
2. Luangphakdy V, Pluhar E, Zachos TA, Boehm C, Liu X, Carlson C, Bechtold JE, D'Alleyrand JC, Muschler GF, Optimizing Soft Tissue Management and Spacer Design in Segmental Bone Defects. ORS, Orlando, Florida, March 5-8, 2016.
3. Muschler GF, Luangphakdy V, D'Alleyrand JCD, Carlson C, Bechtold J, Pluhar GE, Optimizing soft tissue management and spacer design in caprine segmental bone defects, 2016 MHSRS, Orlando/Kissimmee, Florida, August 15-18, 2016.
4. Luangphakdy V, Boehm C, Zachos TA, D'Alleyrand JCD, Carlson C, Bechtold J, Pluhar GE, Muschler GF, Optimizing Soft Tissue Management And Spacer Design In Caprine Segmental Bone Defects. Podium presentation, 2016 TERMIS-AM Meeting, San Diego, California, December 11 – 14, 2016.
5. Muschler G, Luangphakdy V, Piuzzi N, D'Alleyrand JCD, Carlson C, Bechtold J, Pluhar GE, The Masquelet Induced Membrane Technique: Optimizing surgical technique and Spacer Design. Podium presentation, 69th Annual Meeting of The Association of Bone and Joint Surgeons in Austin, Texas, April 5-9, 2017.

7. INVENTIONS, PATENTS AND LICENSES:

Nothing to report

8. REPORTABLE OUTCOMES:

The Chronic Caprine Tibial Defect (CCTD) model 'raises the biological bar" for clinically relevant preclinical assessment of advanced bone regeneration strategies. The CCTD model is now unequivocally the most rigorous, quantitative, and clinically relevant large animal model currently for testing new bone regeneration materials and therapies.

This study highlights that the Chronic Caprine Tibial Defect Model is sensitive variations in: surgical technique (scraping), spacer design (biomaterials, texture, drug delivery) and graft source quality (ACBG fat content and CTP prevalence; and bone marrow aspiration CTP prevalence)

Surgical removal of the inner surface of the IM significantly increased bone regeneration. This finding is important, clinically relevant and immediately translatable into clinical practice to serve wounded warriors as well as civilians.

9. OTHER ACHIEVEMENTS:

ASTM International has initiated a work group led by Dr. Pluhar (ASTM division IV TEMP, F04.44.) to develop the CCTD model as an ASTM Standard Method, recognizing the value of the CCTD model to the bone regeneration community.

10. REFERENCES:

1. Belmont, P.J., B.D. Owens, and A.J. Schoenfeld, *Musculoskeletal Injuries in Iraq and Afghanistan: Epidemiology and Outcomes Following a Decade of War.* J Am Acad Orthop Surg, 2016. 24(6): p. 341-8.
2. Viateau, V., et al., Induction of a barrier membrane to facilitate reconstruction of massive segmental diaphyseal bone defects: an ovine model. Vet Surg, 2006. **35**(5): p. 445-52.
3. Masquelet, A.C. and T. Begue, The concept of induced membrane for reconstruction of long bone defects. Orthop Clin North Am, 2010. **41**(1): p. 27-37; table of contents.
4. Masquelet, A.C., F. Fitoussi, T. Begue, and G.P. Muller, [Reconstruction of the long bones by the induced membrane and spongy autograft]. Ann Chir Plast Esthet, 2000. 45(3): p. 346-53.
5. Giannoudis, P.V., O. Faour, T. Goff, N. Kanakaris, and R. Dimitriou, Masquelet technique for the treatment of bone defects: tips-tricks and future directions. Injury, 2011. 42(6): p. 591-8.
6. Karger, C., T. Kishi, L. Schneider, F. Fitoussi, and A.C. Masquelet, Treatment of posttraumatic bone defects by the induced membrane technique. Orthop Traumatol Surg Res, 2012. 98(1): p. 97-102.
7. Standard Test Method for Automated Colony Forming Unit (CFU) Assays – Image Acquisition and Analysis Method for Enumerating and Characterizing Cells and Colonies in Culture, 2012, ASTM International: West Conshohocken, PA. DOI 10.1520.
8. Nakamoto, C., C. Boehm, J.H. Tao, K. Powell, and G.F. Muschler. Comparison of Bone Marrow Aspiration and Bone Core Biopsy as Methods for Harvest and Assay of Human Connective Tissue Progenitor. in 119th Annual Meeting of The American Orthopaedic Association. June 21-24, 2006. San Antonio, TX.
9. Stekhoven, D.J. and Buehlmann, P. (2012), 'MissForest - nonparametric missing value imputation for mixed-type data', Bioinformatics, 28(1) 2012, 112-118.
10. Miron B. Kursa, Witold R. Rudnicki (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software, 36(11),* p. 1-13.
11. Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. J. Royal. Statistics. Soc. B., Vol. 58, No. 1, 267-288.
12. Venables, W. N. $Ripley, B.D. (2002) Modern Applied Statistics with S. Fourth edition. Springer, New Yor
13. Breiman, L. (2001), *Random Forests*, Machine Learning 45(1), 5-32.

11. APPENDICES: Attach all appendices that contain information that supplements, clarifies or supports the text. Examples include original copies of journal articles, reprints of manuscripts and abstracts, a curriculum vitae, patent applications, study questionnaires, and surveys, etc.

Appendix # 1: Manuscript "The Masquelet induced membrane technique: Optimizing surgical technique and spacer design" Submitted on Nov. 30th, 2016 to Clinical Orthopaedics and Related Research (CORR).

The Masquelet induced membrane technique: Optimizing surgical technique and spacer design

Viviane Luangphakdy, MS, G. Elizabeth Pluhar, DVM, PhD, Nicolás S. Piuzzi, MD, Jean-Claude D'Alleyrand, MD, Cathy S. Carlson, DVM, PhD, Joan E. Bechtold, PhD, Jonathan Forsberg, MD, PhD, George F. Muschler, MD

Abstract

Background: The Masquelet induced membrane technique is a commonly used method for treating segmental bone defects. However, there are no established clinical standards for management of the induced membrane (IM) prior to grafting.

Questions/purposes: Two clinically-based hypotheses were tested in the Chronic Caprine Tibial Defect model: 1) A textured spacer that increases IM surface area will increase bone regeneration; 2) Surgical scraping to remove a thin tissue layer of the inner IM surface will enhance bone formation.

Methods: Thirty-two goats were assigned to four groups: smooth spacer with/without membrane scraping, and textured spacer with/without membrane scraping. During an initial surgery, a defect was created excising bone, periosteum and muscle. Segments were initially stabilized with an intramedullary rod and an antibiotic-impregnated polymethylmethacrylate spacer with smooth or textured surface. 4-weeks later, the spacer was removed and the IM was either scraped or left intact before bone grafting. Bone formation was assessed using x-rays, microCT and histology 12-weeks after grafting.

Results: MicroCT analysis demonstrated significantly greater bone formation in defects with scraped IM compared to defects with intact IM ($p = 0.041$). There were no significant differences in bone formation between textured and smooth spacers.

Conclusion: Scraping the IM surface to remove the innermost layer of the IM increased bone regeneration. A textured spacer that increased the IM surface area had no effect on bone regeneration.

Clinical Relevance: Scraping the IM during the second stage of the Masquelet technique may be a rapid and simple means of improving healing of segmental bone defects, and should be considered in clinical practice.

Level of Evidence: Level I, Pre-clinical Therapeutic Study

Introduction

Segmental bone defects represent a major, unsolved clinical challenge in orthopaedic practice, both in civil and military populations.[4, 5, 20, 23, 25] Bone defects may result from high-energy trauma, infection, tumor resection, or revision surgery.[4, 9, 14, 19, 26] Although acute bone shortening may be performed for the treatment of small defects between 1-3 cm, autogenous cancellous bone graft (ACBG) remains the standard of care for bone defects less than 5 cm in size.[19, 25] For larger bone defects, early grafting with ACBG results in a high failure rate due to graft resorption and lack of consolidation.[14, 19, 25] As a result alternative approaches have been described including distraction osteogenesis, vascularized bone transfer, induced membrane (IM) technique, commonly called the Masquelet technique, and amputation.[3, 4, 14, 19, 25]

Masquelet et al. developed a relatively novel technique to repair large bone defect in two surgical steps: *(1)* In the *first step*, debridement and placement of a polymethylmethacrylate (PMMA) cement spacer loaded with antibiotics with soft tissue coverage if required is done. This first procedure is stabilized using an intramedullary rod or plate fixation; *(2)* In the *second step*, performed four to eight weeks later, the PMMA

spacer is removed and bone graft material is placed within the soft tissue envelope that has formed around the spacer (i.e. the "induced membrane" or "IM"). [15–18]

Large animal models present an instrumental platform in targeting the optimization of strategies for bone regeneration. Standardized and well characterized, reproducible and clinically relevant models, are essential to generate pre-clinical data required to advance therapeutic strategies into clinical practice. This study addresses two technical questions regarding how to optimize the induced membrane technique using a chronic caprine tibial defect (CCTD) model: **1)** The use of a textured spacer that doubles the inner surface area of the IM, during the first stage of the Masquelet technique, will increase bone formation after grafting; and **2)** Surgical scraping to remove a thin (1-2 mm) layer of the inner surface of IM immediately adjacent to the spacer will enhance bone formation after bone grafting.

Materials and Methods

Thirty-two skeletally mature female goats, age 5 ± 1 years (mean ± SD) weighing 50 ± 4 kg underwent the CCTD procedure and were randomly assigned to four groups (8 animals/group) using a 2 x 2 test matrix (factorial design). These groups included: (1) smooth spacer and intact IM; (2) smooth spacer and scraped IM; (3) textured spacer and intact IM; and (4) textured spacer and scraped IM. Study animals were cared for in accordance with the principles of the Guide for the Care and Use of Laboratory Animals[22] after approval from the Cleveland Clinic Institutional Animal Care and Use Committee (IACUC # 2013-1021) and the Animal Care and Use Review Office of US Army Medical Research and Materiel Command (OR # 120082).

Chronic Caprine tibial defect (CCTD) model: detailed methods

Prior to each surgery, goats were given perioperative analgesia using staged application of transdermal fentanyl patches (each 50 µg/hr). Medetomidine (0.025 mg/kg intramuscular) and ketamine (0.5−1.0 mg/kg intramuscular) were given for anesthetic induction to place an endotracheal tube and anesthesia was maintained with isoflurane 1.0−2.0% in oxygen. Cefazolin, 1 g IV, was given prophylactically just prior to and at the end of surgery. Morphine (0.1 mg/kg), diluted in 0.9% sterile saline to a volume of 0.13 ml/kg, was given for analgesia in the epidural space after induction.

The first surgery consisted on two procedures: 1) creation of the critical bone defect, and 2) "spacer procedure" or "stage 1" of the Masquelet technique. During the first surgery, a medial approach was made to the left tibia. A 5 cm mid-diaphyseal osteoperiosteal segment was removed beginning 6.5 cm above the medial malleolus. Additional 2-cm segments of periosteum were removed on both side of the defect, as well as 10 gm (~10 cm^3) of skeletal muscle surrounding the defect site. The tibia was stabilized with an 8-mm diameter, 185-mm long intramedullary interlocking nail (Innovative Animal Products, Rochester, MN) and the defect was filled with a polymethylmethacrylate cement spacer impregnated with vancomycin (0.5 g) and tobramycin (0.5 g) (Simplex P polymer, Stryker, Mahwah, NJ) (diameter 2.1 cm). As indicated previously, 16 goats were assigned to the smooth spacer group (**Figure 1A**) and the remaining 16 goats were assigned the textured spacer group (**Figure 1D**) during this first surgery. The textured spacer was fabricated with 2mm thick and 2mm deep linear groves, which doubled the effective surface area of the spacer and the resulting induced membrane (See **Figure 1D** to **1F**).

The second surgery, ("Grafting procedure" or "Stage 2" of the Masquelet technique) was performed four weeks later. The PMMA spacer was removed through a medial approach using a "bomb bay door" opening technique (See **Figures 1B** and **1E**). In the scraped IM group the inner layer of the IM was removed with a curette (n = 16 goats) comprising the loosely adherent zone of foreign body reaction around the spacer, exposing a more fibrous surface of dense connective tissue within the IM. In the intact IM group, the inner IM layer was left intact (n = 16 goats). (**Figure 1B, C, E & F**) Subsequently the defect in all cases was filled with 10-12 cm^3 of ACBG harvested from the sternum. The induced membrane and skin were separately closed with 3-0 polydioxanone and nylon, respectively.

Post-operative pain management was administered using transdermal fentanyl patches (100 µg/hr x 5 days) and phenylbutazone (1 g P.O.). Animals were allowed to be freely mobile and fully weight bearing immediately on recovery from anesthesia. They were housed in individual pens with free access to food and water. The goats were checked twice daily for mentation/ attitude, ability to ambulate, willingness to bear

weight on the operated limb, food and water consumption, respiratory rate, and inflammation at the surgical site.

The goats were euthanized twelve weeks after the second surgery (stage 2 of the Masquelet technique), and the entire left tibia was harvested, preserving the periosteum and fibrous tissue surrounding the defects. Samples were fixed in 10% neutral buffered formalin for 48 hours and then transferred to 70% ethanol (EtOH).

MicroCT processing and analysis

High-resolution microCT was used as the primary outcome measure to characterize the amount and distribution of bone within each defect. Prior to scanning, the nail and screws were carefully removed and replaced with a radiolucent rod and cross-pins to maintain the original length and axial and rotational alignment. After centering in the scanner (Inveon microCT scanner, Siemens Medical Solutions, Knoxville, TN), 441 projection images were acquired at 0.5° increments at 39-µm voxel resolution (80 kVp; 500 µA). Data were reconstructed into 3D volumes using a modified tent-Feldkamp algorithm and calibrated to mg/cm^3 of HA using an air/water/hydroxyapatite phantom scanned under the same conditions. Bone threshold was set at 1300 mg HA/cm^3 (747 HU). The analyzed volume of interest (VOI) included the 5 cm defect region plus 1 cm of bone proximal and distal to the defect. Each specimen was analyzed with a segmentation software developed in-house, in which a 3-dimensional cylindrical "defect template" volume 45mm in diameter and 70 mm in length, was manually positioned to define the boundaries of the VOI. Primary outcome was defined as total bone volume (tBV) in the central 2.5 cm region of the defect, which is the most challenging region to heal. Secondary outcomes were radial percent bone volume (%BV), a summation of circumferential bone volume over the length of the 5cm defect, and angle-moment of inertia in the defect site. Summary data of the pattern and extent of mineralization in the defects in each group were illustrated by projecting %BV versus radial position using a 2D color map ranging from 0% (purple) to 60% (red). The x-axis indicated distance from the center of the medullary canal to the periosteal surface (range = 0-29 mm). The y-axis represented the position in the long axis of the bone (range 0 - 70 mm) including the 5 cm defect plus 1 cm proximal and distal. Mean Angle-moment of inertia for each group was also plotted using a 2D color plot color map ranging 0 (dark blue) to 7000 (red) in Hounsfield units (HU)*mm^{2}plotted about the bone circumference in the x-axis and the long bone axis position in the y-axis.

Radiographic Analysis

Fluoroscopic imaging of the tibiae, anterior-posterior (AP) and medio-lateral (ML) projections, were performed after: the spacer procedure (week 0), the graft procedure (week 4), follow-up (week 8)(Figure 4). Radiographs were obtained after euthanasia (after soft tissues were dissected) 12 weeks after the grafting procedure. The resulting images were ranked from 1 (greatest bone healing) to 20 (no healing) by two independent investigators (senior orthopaedic surgeon and junior orthopaedic surgeon) who were blinded to treatment allocation. The highest-ranking sites revealed bony bridging (bone extending the length of the defect with no discontinuity) of all four cortices, followed by bony bridging of three, two, one, or none of the cortices. Among samples that had the same number of bony bridges, the ranking was based on the subjective amount of bone observed.

Histology Analysis

Following microCT data collection, histology and histomorphometry were performed. Each fixed tibia was immersed in dilute HCl (~1 week) and then transferred to a 10% EDTA solution until completely decalcified. Decalcified specimens were embedded in paraffin, trimmed to 7 cm to include the 5 cm defect and 1 cm at either end. The specimens were compared with the corresponding faxitron radiographs to ensure that the correct tissue area was examined. The specimens were then divided into four 1.75 cm long segments that were then cut in the craniocaudal direction on the mid-sagittal plane to yield medial and lateral halves. Medial halves (each including the anterior and posterior cortices) were processed in paraffin (four blocks/specimen), sectioned at 5-µm, and stained with hematoxylin and eosin (H&E) and Masson's trichrome. For the histomorphometric analysis, total area of bone in the sections from the two central 1.75 cm blocks (total of 3.5 cm) was measured by tracing the perimeter of each individual focus/area of bone tissue in the section in both anterior and posterior

cortices (mm^2) using SPOT basic histomorphometry software (SPOT Imaging, Sterling Heights, MI) and summing the results for each animal.

Statistical Analysis

A power analysis using previous data collected with this animal model was performed to estimate the sample size required for detecting a difference in new bone volume parameter. Using the traditional alpha set to 0.05, 16 goats (8 animals per group) provides a power of 0.78 to detect a change from 20% bone volume to 30%. This 10% difference in a 12 cm^3 defect is 1,2 cm^3, and was considered to be clinically significant.
All statistical tests were performed using general linear models using one-sided tests at a significance level of 0.05 with SAS 9.4 software (SAS Institute, Cary, NC). The response variable was total bone volume (tBV in mm^3) calculated from the microCT scans in the central 2.5 cm of the defect and the treatment factors were scraped or non-scraped IM and textured or smooth spacers. All data were expressed as mean ± standard error.

Results

Three goats were excluded due to complications. Two belonged to the intact/smooth group and were excluded due to Staphylococcus epidermidis infection (n=1) or caseous lymphadenitis (n=1); one goat in the scraped/smooth group was excluded due to anesthetic complication during stage 2 surgery (n=1).

Bone formation assessed by MicroCT

Mean radial tBV in the central 2.5 cm of the defect was significantly greater in the IM scraped group (mean +/- standard error = 1861.0 ± 351.3 mm^3) vs. non-scraped IM (930.3 ± 390.3 mm3) ($p = 0.041$). However, there was no trend or significant difference between the smooth or textured spacer groups ($p = 0.475$) (**Figure 2**). The mean tBV in the central 2.5 cm of the defects was 1413.8 ± 363.3 mm^3 for the smooth membrane group and 1377.5 ± 377.5 mm^3 for the textured membrane group. These data are graphically presented using a heat map in Figure 3. Mean percent bone volume in **Figure 3** illustrates that there is greater bone formation near the osteotomy sites. The angle-moment of inertia plots illustrate that bone tended to form along the posterior and medial aspect of the defect with little bone formation laterally in the IM scraped group while in the intact IM group, new bone formation was found posteriorly. Textured and smooth spacer groups had comparable new bone formation; with bone preferentially forming posteriorly in both groups as shown in the angle moment plots.

Postmortem radiograph ranking

Overall, the scraped IM group ranked better than the non-scraped IM group (**Supplemental Figure S1**). Ten of the top twelve radiographs belonged to the scraped IM group. The radiographic ranking did not reveal any differences between smooth and textured spacer groups.

Histology

Histological assessment of the goat tibia samples identified no evidence of inflammation in any of the sections 12 weeks after grafting. The majority of the bone that was present in the sections was composed of cancellous woven bone. There was a large amount of variation in the amount of bone present in the defects (**Figure 5**). Defects containing little or no bone were filled primarily with fibrous connective tissue. The most robust bone regeneration occurred in the scraped IM group (mean total bone area = 244.1 ± 67.8 mm^2) compared to the intact IM group (177.0 ± 74.3 mm^2), which is consistent with the microCT data. The histomorphometry data also showed that textured spacer group (mean total bone area = 296.9 ± 94.3 mm^2) had a larger area of bone in the defect than smooth spacer group (136.7 ± 30.4 mm^2). No statistically significant difference was found with histomorphometry assessment between groups.

Discussion

The Masquelet technique is an effective two-stage method to enhance bone reconstruction by creating a vascularized soft tissue envelope prior to grafting of bone defects.[2, 4, 6, 11, 15, 16, 18, 25, 28] Optimizing the

Masquelet IM method and characterization of the biological features of the IM has significant potential to enhance the clinical care of patients who require bone regeneration procedures to treat critical segmental bone defects. The main finding of this study was that scraping the inner surface of the IM during the second stage of the Masquelet technique was associated with improved bone formation that nearly doubled the mean amount of bone formation compared to the control group.

The IM is composed of two portions: a thin glistening inner surface, comprised of protein exudate and mild foreign body reaction, and an outer part made of fibroblasts, myofibroblasts, and collagen.[24] This inner surface has been described previously as an organized richly vascularized pseudo-synovial membrane, that promotes the revascularization of the graft that is contained within the membrane and prevents its resorption.[13, 16, 24, 29] Many biological properties have been attributed to the IM, including expression of bone morphogenetic protein-2 (BMP-2), transforming growth factor-b, vascular endothelial growth factor (VEGF), von Willebrand factor, interleukin 6, interleukin 8, type 1 collagen, and stroma-derived factor 1, and it also has been described as a source of stem/progenitor cells.[1, 7, 8, 12, 24, 29, 30] Aho et al. reported that the IM osteogenetic capability with respect to VEGF expression and vascularity was achieved one month after spacer implantation.[1]

Although the Masquelet technique is used worldwide[10] and a significant amount of basic and clinical research has been done on the IM technique, there are no established surgical standards to manage the IM inner surface prior to bone grafting. Removal of the glistening inner layer of the IM induces bleeding from a healthy vascular bed, while preserving the mechanical function of the rest of the fibrous envelope of the IM. This technical surgical modification may represent a rapid and simple means of enhancing the biological environment of the tissue bed. Further investigation into the nature of the IM including exploring differences in resident cell populations and difference in gene expression between superficial and deep zones in the IM may help define a mechanism for this effect.

A textured spacer has been shown to be associated with more developed membrane synovial-like metaplasia and villous hyperplasia,[16] but the effect of spacer texture on bone healing is not well known. We tested a ribbed spacer that doubled the IM surface area to try to enhance bone healing, however we did not find a difference in bone volume between the two spacer designs. However, there are many other texture options that may be considered before texture should be discarded as a possible opportunity for improvement. Other variables in the optimization of the Masquelet Technique that may be considered are: spacer bulk materials (e.g. other polymers besides PMMA), variation in time allotted to each stage, delivery of growth factors using a spacer, type and amount of grafting material, and supplementation with cell-based therapies.[4, 16, 20, 27, 29]

Our study has two primary limitations. While the process of scraping using the CCTD model was associated with a near doubling in bone formation when ACBG was used, the variation in the data was high and scraping did not automatically translate into clinical success. The risk of an alpha error (finding an effect by chance) remains close to 4%. Given the simplicity of this method and the very low risk or effort involved, we believe that it is appropriate to test this surgical technique variation in appropriately designed human clinical settings. It is also appropriate for the scraping method to be further assessed in settings where other animal models using other graft materials are used, to further test and confirm the value of this method in a larger range of conditions. Advancement in surgical clinical care of bone defects has been limited by the fact that available large animal models, generally present acute bone defects in young healthy animals, that heal reliably with many existing therapies, creating a "ceiling effect".[21] The CCTD model addresses the latter limitation and provides a robust model that "raises the bar" for rigorous assessment of bone grafting strategies. Even ACGV achieved a mean of only 4 cm^3 of new bone in a 12 cm^3 defect site. Moreover, this model includes the biological features of muscle injury, loss of periostium, bone marrow reaming and local scar formation, which are missing in traditional acute large animal models. Therefore these features better model the challenging biological environment where advancement needs to be made and documented. The most sensitive test for new bone formation is in the center-most 2.5 cm of the bone defect, eliminating reactive bone formation that may occur independent from the grafting strategy. Furthermore histology was used as a secondary outcome to evaluate and confirm the quality of the new bone tissue formed.

Conclusion

When using the Masquelet technique, scraping to remove the inner surface of the induced membrane, prior to bone grafting, may improve clinical bone regeneration. The scraping of the inner surface of the induced membrane increases bleeding from a healthy vascular fibrous tissue bed and removes a layer of tissue associated with a foreign body reaction, while preserving the mechanical function of the IM as a fibrous envelope.

This study highlights the opportunity of optimizing surgical techniques to improve healing of segmental bone defects using a rigorous chronic caprine tibial defect model for preclinical assessment of bone regeneration materials and strategies. The findings from this study can immediately be translated to the clinical care of civilians and military patients.

Acknowledgments

We gratefully acknowledge Cynthia Boehm, Wesley Bova, Maha Qadam, Venkata Mantripragada and Terry Zachos for their technical assistance during surgery; Dr. Kimerly Powell and Dr. Justin Jeffery for help with microCT processing and analysis; Ferenc Toth, Elizabeth Marchant and Channing Bancroft for help with histology processing and analysis; Xiaobo Liu for assistance with statistical analysis.

References

1. Aho O, Lehenkari P, Ristiniemi J, Lehtonen S, Risteli J, Leskela H-V. The Mechanism of Action of Induced Membranes in Bone Repair. *J. Bone Jt. Surg.* 2013;95–A:597–604.

2. Apard T, Bigorre N, Cronier P, Duteille F, Bizot P, Massin P. Two-stage reconstruction of post-traumatic segmental tibia bone loss with nailing. *Orthop. Traumatol. Surg. Res.* 2010;96:549–553.

3. Ashman O, Phillips AM. Treatment of non-unions with bone defects: Which option and why? *Injury*. 2013;44:S43–S45.

4. Aurégan JC, Bégué T. Induced membrane for treatment of critical sized bone defect: A review of experimental and clinical experiences. *Int. Orthop*. 2014;38:1971–1978.

5. Brown K V., Guthrie HC, Ramasamy A, Kendrew JM, Clasper J. Modern military surgery: Lessons from Iraq and Afghanistan. *Bone Joint J.* 2012;94–B:536–543.

6. Chong K-W, Yi-Loong Woon C, Wong M. Induced Membranes—A Staged Technique of Bone-Grafting for Segmental Bone Loss Surgical Surgical Technique. *J. Bone Jt. Surg.* 2011;93:85–91.

7. Christou C, Oliver RA, Yu Y, Walsh WR. The Masquelet Technique for Membrane Induction and the Healing of Ovine Critical Sized Segmental Defects. *PLoS One*. 2014:1–19.

8. Cuthbert RJ, Churchman SM, Tan HB, McGonagle D, Jones E, Giannoudis P V. Induced periosteum a complex cellular scaffold for the treatment of large bone defects. *Bone*. 2013;57:484–492.

9. Dumic-Cule I, Pecina M, Jelic M, Jankolija M, Popek I, Grgurevic L, Vukicevic S. Biological aspects of segmental bone defects management. *Int. Orthop*. 2015;39:1005–1011.

10. Giannoudis P V. Has the Induced Membrane Technique Revolutionalized the Treatment of Bone Defects ? *Tech. Orthop.* 2016;31:2016.

11. Giannoudis P V, Faour O, Goff T, Kanakaris N, Dimitriou R. Masquelet technique for the treatment of bone defects: tips-tricks and future directions. *Injury*. 2011;42:591–598.

12. Gouron R, Petit L, Boudot C, Six I, Brazier M, Kamel S, Mentaverri R. Osteoclasts and their precursors are present in the induced-membrane during bone reconstruction using the Masquelet technique. *J. Tissue Eng. Regen. Med.* 2016.

13. Klaue K, Knothe U, Anton C, Pfluger H, Stoddart M, Masquelet AC, Perren SM. Research Ne s from AO Bone regeneration in long-bone efects : tissue compartmentalisation ? In vivo stu on bone efects in sheep. *Injury*. 2009;40S4:S95–S102.

14. Lasanianos NG, Kanakaris NK, Giannoudis P V. Current management of long bone large segmental defects. *Orthop. Trauma*. 2010;24:149–163.

15. Masquelet AC. The Evolution of the Induced Membrane Technique : Current Status and Future Directions. *Tech. Orthop.* 2016;31:3–8.

16. Masquelet AC, Begue T. The concept of induced membrane for reconstruction of long bone defects. *Orthop. Clin. North Am.* 2010;41:27–37; table of contents.
17. Masquelet AC, Fitoussi F, Begue T, Muller GP. [Reconstruction of the long bones by the induced membrane and spongy autograft]. *Ann. Chir. Plast. esthétique*. 2000;45:346–53.
18. Masquelet AC, Obert L. La technique de la membrane induite pour les pertes de substance osseuse de la main et du poignet. Induced membrane technique for bone defects in the hand and wrist. *Chir. Main*. 2010;29S:S221–S224.
19. Mauffrey C, Barlow BT, Smith W. Management of Segmental Bone Defects. *J Am Acad Orthop Surg*. 2015;23:143–153.
20. Mauffrey C, Hake ME, Chadayammuri V, Masquelet A-C. Reconstruction of Long Bone Infections Using the Induced Membrane Technique: Tips and Tricks. *J. Orthop. Trauma*. 2015;30:188–193.
21. Muschler GF, Raut VP, Patterson TE, Wenke JC, Hollinger JO. The design and use of animal models for translational research in bone tissue engineering and regenerative medicine. *Tissue Eng. Part B. Rev.* 2010;16:123–45.
22. National Research Council. *Guide for the care and use of laboratory animals*. 8th Editio. Washingtong, DC: National Academies Press; 2011.
23. Owens BD, Kragh JF, Macaitis J, Svoboda SJ, Wenke JC. Characterization of extremity wounds in Operation Iraqi Freedom and Operation Enduring Freedom. *J. Orthop. Trauma*. 2007;21:254–257.
24. Pelissier P, Masquelet AC, Bareille R, Mathoulin Pelissier S, Amedee J. Induced membranes secrete growth factors including vascular and osteoinductive factors and could stimulate bone regeneration. *J. Orthop. Res.* 2004;22:73–79.
25. Pipitone PS, Rehman S. Management of Traumatic Bone Loss in the Lower Extremity. *Orthop. Clin. North Am.* 2014;45:469–482.
26. Pneumaticos SG, Triantafyllopoulos GK, Basdra EK, Papavassiliou AG. Segmental bone defects: From cellular and molecular pathways to the development of novel biological treatments. *J. Cell. Mol. Med.* 2010;14:2561–2569.
27. Pountos I, Panteli M, Jones E, Giannoudis P V. How the Induced Membrane Contributes to Bone Repair : *Tech. Orthop.* 2016;31:9–13.
28. Ronga M, Ferraro S, Fagetti A, Cherubino M, Valdatta L, Cherubino P. Masquelet technique for the treatment of a severe acute tibial bone loss. *Injury*. 2014;45:S111–S115.
29. Viateau V, Bensidhoum M, Guillemin G, Petite H, Hannouche D, Fani A, Pelissier P. Use of the Induced Membrane Technique for Bone Tissue Engineering Purposes : Animal Studies. *Orthop Clin North Am*. 2010;41:49–56.
30. Viateau V, Guillemin G, Bousson V, Oudina K, Hannouche D, Sedel L, Logeart-Avramoglou D, Petite H. Long-Bone Critical-Size Defects Treated with Tissue-Engineered Grafts : A Study on Sheep. *J. Orthop. Res.* 2007;25:741–9.

Figure 1 – Surgery pictures showing placement of a smooth spacer (**1A**) and textured spacer (**1D**) in the defect site during the "spacer-procedure" (Stage 1, Masquelet technique). Four week later during the "grafting procedure" (Stage 2, Masquelet technique) the induced membrane is appreciated after smooth spacer (**1B**) and textured spacer (**1E**) removal. Scraping of the inner layer of the induced membrane with a curette (yellow arrow) was done both in half of the goat from the smooth spacer group (**1C**) and textured spacer group (**1F**).

Figure 2 – Total bone volume plots, based on microCT analyses, of the central 2.5 cm of the defect and the entire 5 cm defect illustrating the effect of scraping the IM (**2A**) and the effect of the spacer texture (**2B**)

Figure 3 – MicroCT plots illustrating the differences in bone formation and distribution. **3A**) A radial color plot illustrates the bone formation (Bone volume %, BV%) as the summation of the radial location relative to vertical position in the defect. More BV% (red/yellow color) is appreciated in the scraped group compared to the intact IM group. 3B) A moment-angle plot illustrates that most bone formed on the posterior aspect of this defect. The proximal and distal borders of the defect are marked by dashed lines.

Figure 4 – Representative anterior-posterior and lateral views of fluoroscopic images taken during the experiment: **4A**) after the "spacer-procedure" (Stage 1) showing the PMMA spacer in the defect site; **4B**) after the "graft procedure" (Stage 2) 4 weeks later showing ACBG in the defect; and **4C**) at the time of euthanasia (12 weeks after Stage 2) showing new bone formation in the defect.

Figure 5 – Example of postmortem radiographs (anterior-posterior and lateral views), MicroCT scan 3D reconstruction, % Bone Volume (BV) plot versus summation of radial position, and Masson's Trichrome stained histology slides (pink staining= bone formation) for two representative goats: one showing robust new bone formation (**6A, 6B, 6C, 6D**); and one showing minimal new bone formation (**6E, 6F, 6G, 6H**).

Figure S1 – Postmortem radiographs (anterior-posterior and lateral views) of the tibia from each goat used for ranking. Higher rank number indicates greater bone formation in the tibial defect (1 = complete bone healing to 29 = no healing).

Appendix # 2: Predictive analysis of the Goat data using R software

goat_IM_analysis.R

by John Tra, PhD

```
#################################################################################################
###

#  load libraries
library(knitr)
library(Boruta)

## Loading required package: ranger

library(randomForest);

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ranger':
##
##     importance

library(missForest)

## Warning: package 'missForest' was built under R version 3.3.2

## Loading required package: foreach

## Loading required package: itertools

## Warning: package 'itertools' was built under R version 3.3.2
```

```
## Loading required package: iterators
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library(e1071)
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(nnet)
```

```
## Warning: package 'nnet' was built under R version 3.3.2
```

```
#Load goat data
goat_IM <- read.csv("~/Data Analysis for collaborators/Muschler/goatfinal.csv")
str(goat_IM)
```

```
## 'data.frame':    32 obs. of  70 variables:
##  $ id                              : int  1 3 6 8 9 14 15 17 2 4 ...
##  $ Goat.ID                         : Factor w/ 32 levels "14G1","14G10",..: 1
10 13 15 16 6 7 17 9 11 ...
##  $ weight                          : int  53 48 54 54 41 59 38 44 61 53 ...
##  $ Pre.Tx.Surgeon                  : Factor w/ 4 levels "GFM","JCDA","LP",..:
1 1 2 2 2 3 2 3 3 3 ...
##  $ Tx.Surgeon                      : Factor w/ 4 levels "GFM","JCDA","LP",..:
3 3 1 2 2 1 2 3 3 1 ...
##  $ ACBGBMA.Surgeon                 : Factor w/ 4 levels "GFM","JCDA","LP",..:
```

```
3 1 3 1 1 1 2 4 1 2 ...
##  $ scraping                          : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ Spacer.texture                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CT.2.5                            : num  132 135 819 4021 1272 ...
##  $ CT.5                              : num  682 1237 1940 9534 3364 ...
##  $ Xray.Rank                         : int  26 23 16 5 14 27 22 NA 15 6 ...
##  $ Tib_Histo_Bone.Area..3.5          : num  11.1 33.2 109 358 114.7 ...
##  $ IM__Histo_Vascularity             : int  1 2 1 1 1 1 1 1 1 1 ...
##  $ IM_Histo_Fibrosis                 : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ IM_Histo_inflammatory_Cellularity: int  0 0 0 0 1 0 0 0 1 0 ...
##  $ IM_Histo_Thickness                : num  3.06 5.48 4 8.96 5.05 ...
##  $ IM.membrane._.weight              : num  0.0975 0.12 0.1275 0.185 0.135 ...
##  $ IM_Inner.Mean.cellularity         : num  24.13 9.83 55.42 25.83 82.5 ...
##  $ Expanded.IM_inner.cells           : num  1340 266 676 554 7036 ...
##  $ IM_Outer.Mean.cellularity         : num  19.94 6.5 7.5 8.25 23.92 ...
##  $ Expanded.IM_outer.cells           : num  1123 NA NA 582 10209 ...
##  $ IM_inner_RunX2                    : num  18.6 11.2 12.5 29.8 13.5 ...
##  $ inner_Osterix                     : num  0.53 -0.15 1.65 1.91 -1.99 3.01 -0.
5 29.3 -0.39 1.45 ...
##  $ inner_PPARG                       : num  22.52 13.29 6.66 4.79 9.01 ...
##  $ inner_OCT4                        : num  -3.43 -5.09 -8.39 -4 -1.9 ...
##  $ inner_Sox2                        : num  38.8 -10.4 73.9 1347.8 564.3 ...
##  $ inner_NANOG                       : num  -17.2 -10.9 -15 -10.2 -26.7 ...
##  $ inner_ALPL                        : num  1.52 0.36 1.34 2.78 0.08 3.73 0.05
8.84 1.27 1.79 ...
##  $ inner_EGFR                        : num  3.52 4.32 2.03 3.78 0.21 ...
##  $ inner_VWF                         : num  4.35 2.91 2.02 1.51 2.66 ...
##  $ inner_PDGFB                       : num  7.05 3.53 4.47 3.11 4.45 ...
##  $ inner_TGFB1                       : num  26.4 15.3 28.4 15.5 16 ...
##  $ inner_BMP2                        : num  2.01 2.97 3.78 8.98 1.96 ...
##  $ inner_BMP6                        : num  2.47 4.09 2.21 89.66 2.12 ...
##  $ inner_Col1A1                      : num  364.58 3451.96 184.98 -2.65 181.04
...
##  $ inner_Col2A1                      : num  -8.05 -4.4 -12.38 -4.37 -9.25 ...
##  $ inner_Col10A1                     : num  -17.88 15.3 16.23 9.11 -4.57 ...
##  $ inner_GLA                         : num  -1.52 -1.57 -0.43 75.82 -0.53 ...
##  $ inner_IBSP                        : num  0.16 3.28 4.59 10.27 1.89 ...
##  $ inner_PTHR                        : num  4.61 5.2 2.44 1919.28 3.14 ...
##  $ outer_RunX2                       : num  10.8 9.54 12.78 17.37 13.81 ...
##  $ outer_Osterix                     : num  0.06 -0.46 4.65 1.26 1.93 ...
##  $ outer_PPARG                       : num  19 14.4 12 16.1 12.5 ...
##  $ outer_OCT4                        : num  -2.78 -1.59 -0.39 4.22 -2.05 ...
##  $ outer_Sox2                        : num  249 4342 169 152 20653 ...
##  $ outer_NANOG                       : num  -17.91 -7.61 -5.64 -2.91 -8.74 ...
##  $ outer_ALPL                        : num  1.3 -1.31 2.3 1.77 1.38 -0.31 -0.26
377 -0.05 1.3 ...
##  $ outer_EGFR                        : num  4.51 6.67 5.75 8.62 4.35 ...
##  $ outer_VWF                         : num  3.79 3.07 3.36 7.15 2.94 ...
##  $ outer_PDGFB                       : num  6.65 1827.2 5.29 6.58 4.72 ...
##  $ outer_TGFB1                       : num  23.9 12.1 16.9 25.5 25 ...
```

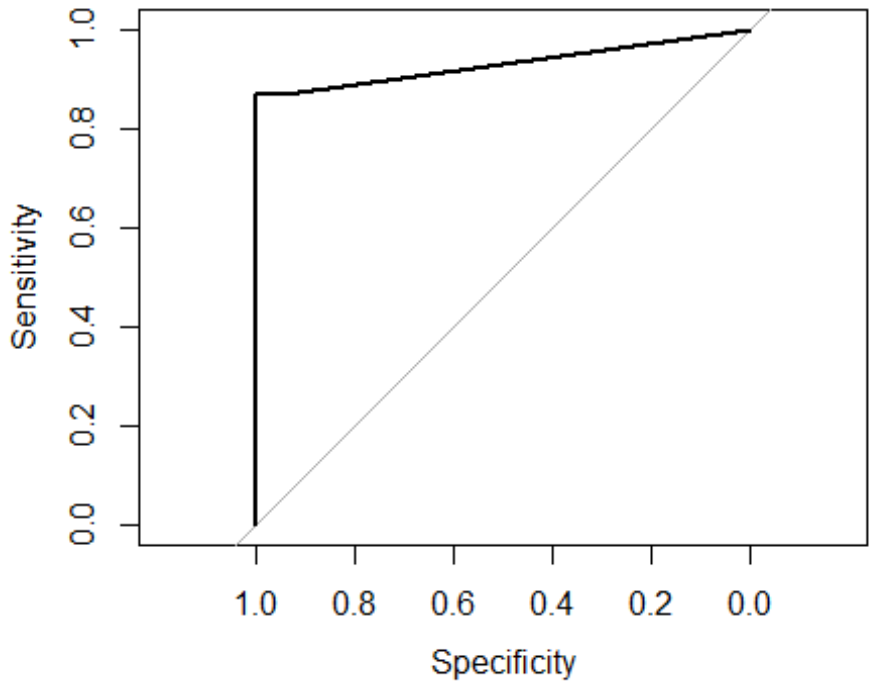
```
##  $ outer_BMP2                          : num  2.88 2.7 4.22 4.26 2.97 ...
##  $ outer_BMP6                          : num  3.37 4.9 6.23 8.18 3.08 ...
##  $ outer_Col1A1                        : num  532 410 635 1354 1757 ...
##  $ outer_Col2A1                        : num  -3.57 -5.8 -2.74 2.4 -2.87 ...
##  $ outer_Col10A1                       : num  -11.42 208.74 5927.69 -1.61 36.57 .
..
##  $ outer_GLA                           : num  0.8 -1.04 0.15 -1.09 0.18 ...
##  $ outer_IBSP                          : num  0.04 -1.59 5.02 2.8 1.17 ...
##  $ outer_PTHR                          : num  4.8 6.09 10.09 10.78 6.16 ...
##  $ BMA.cellularity                     : num  39.6 94.7 18.8 30.5 47.9 ...
##  $ BMA.CTP.Prevalence                  : num  0 32.4 0.77 4.63 59.66 ...
##  $ MS.cellularity                      : num  128 155 220 291 129 ...
##  $ MS.CTP.Prevalence                   : num  3.28 19.69 6.56 50.87 59.07 ...
##  $ TS.cellularity                      : num  76.2 70.9 63.9 100 100 ...
##  $ TS.CTP.Prevalence                   : num  139.5 0 164.1 57.4 188.7 ...
##  $ Graft_Histo_cartilage               : num  17.5 0 0 0 0 ...
##  $ Graft_Histo_.bone                   : num  18.6 24 29.6 27.8 20.4 ...
##  $ Graft_Histo_hematopoetic.marrow     : num  63.9 75.8 70.4 72.2 79.6 ...
##  $ Graft_Histo_adipose.tissue          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Graft_Histo_vasculature.area        : num  0 0.18 0 0 0 0 0 0 0 0 ...
```

```
# convert scraping and spacer texture to factor variable

goat_IM$scraping <- as.factor(goat_IM$scraping)
goat_IM$Spacer.texture<-as.factor(goat_IM$Spacer.texture)

#############################################################################################
#                   imputing for the missing data using missforest
#############################################################################################

goatmiss<- missForest(goat_IM, xtrue = goat_IM , verbose = TRUE,replace=TRUE)
```

```
##   missForest iteration 1 in progress...done!
##     error(s): NA 0
##     estimated error(s): 3.084635e-05 0
##     difference(s): 6.394375e-12 0
##     time: 0.25 seconds
##
##   missForest iteration 2 in progress...done!
##     error(s): NA 0
##     estimated error(s): 2.986406e-05 0
##     difference(s): 3.148973e-12 0
##     time: 0.25 seconds
##
##   missForest iteration 3 in progress...done!
##     error(s): NA 0
##     estimated error(s): 3.045528e-05 0
##     difference(s): 3.318771e-13 0
##     time: 0.23 seconds
##
##   missForest iteration 4 in progress...done!
```

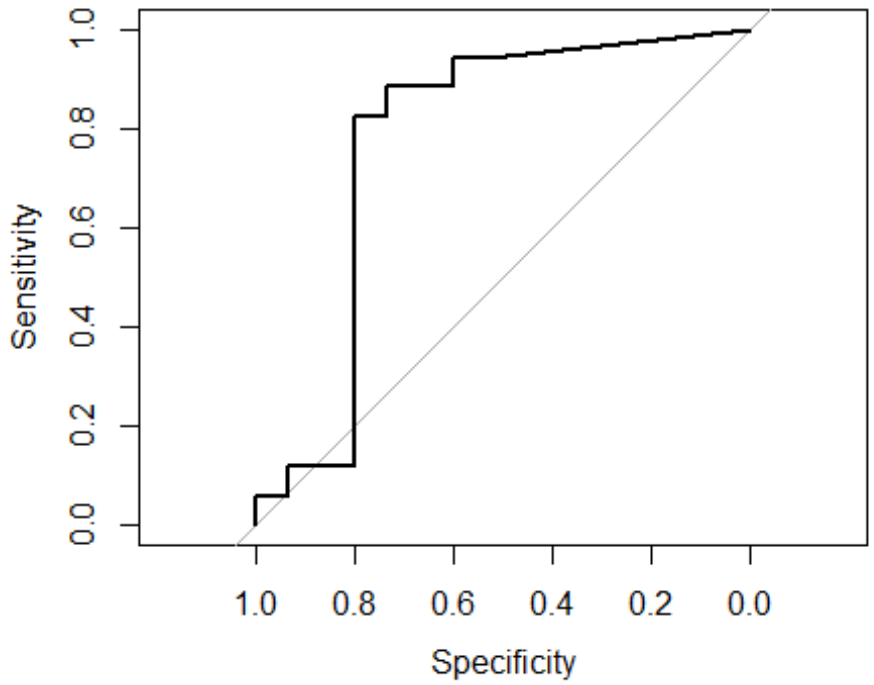
```
##     error(s): NA 0
##     estimated error(s): 2.923674e-05 0
##     difference(s): 1.530613e-12 0
##     time: 0.28 seconds

goat_IM<- as.data.frame(goatmiss$ximp)

#############################################################################
#                   partition the data between training and test sets
#############################################################################


inTraining <- createDataPartition(goat_IM$id, p=0.50, list = FALSE) #split 50%:50%
training <- goat_IM[inTraining, ]
testing <- goat_IM[-inTraining, ]
dim(training)

## [1] 16 70

dim(testing)

## [1] 16 70

#############################################################################################
########################################
# using spacer texture as the response variable, build a model using all the variables to
find the effects on spacer texture
#############################################################################################
########################################

#############################################################################
#   SPACER TEXTURE as a response varibale to build model 1
#############################################################################

#apply Naive Bayes algorithm
##############################


nb <-naiveBayes(Spacer.texture~., data=training,laplace = 0,na.action = na.pass)
summary(nb)

##         Length Class  Mode
## apriori  2     table  numeric
## tables  69     -none- list
## levels   2     -none- character
## call     4     -none- call

#make predictions
pred<- predict(nb,goat_IM, type="raw", Threshold =0.001,eps=0)
table(predict(nb, goat_IM), goat_IM[,8])

##
##      0  1
```

```
##   0 16  2
##   1  1 13

#How good is our prediction?
#########################################################################


#########################################################################
x_test <- goat_IM[,-c(8)]
y_test <- goat_IM[,8]

predictions <- predict(nb, x_test, type="raw")

mroc <-roc(goat_IM[,8], predictions[,1], plot=T)
```




```
coords(mroc, .5, "threshold", ret=c("sensitivity", "specificity", "ppv", "npv"))

## sensitivity specificity         ppv         npv
##   0.8666667   0.9411765   0.9285714   0.8888889

coords(mroc, .95, "threshold", ret=c("sensitivity", "specificity", "ppv", "npv"))

## sensitivity specificity         ppv         npv
##   0.8666667   0.9411765   0.9285714   0.8888889

#########################################################################################
#############
#                          Importance of MB Scraping as a response variable used to bu
ild a model 2
#########################################################################################
#############
```

```
x_test <- goat_IM[,-c(7)]
y_test <- goat_IM[,7]

predictions <- predict(nb, x_test, type="raw")
nb1 <-naiveBayes(scraping~., data=training,laplace = 0,na.action = na.pass)
summary(nb1)

##         Length Class  Mode
## apriori  2     table  numeric
## tables  69     -none- list
## levels   2     -none- character
## call     4     -none- call

pred1<- predict(nb1,goat_IM, type="raw", Threshold =0.001,eps=0)
table(predict(nb1, goat_IM), goat_IM[,7])

##
##      0  1
##   0 12  4
##   1  3 13

x_test <- goat_IM[,-c(7)]
y_test <- goat_IM[,7]

predictions <- predict(nb1, x_test, type="raw")

#########################################################################################
#################
#                                                    Model sensitivity 2
#########################################################################################
#################

mroc <-roc(goat_IM[,7], predictions[,1], plot=T)
```




```
coords(mroc, .5, "threshold", ret=c("sensitivity", "specificity", "ppv", "npv"))

## sensitivity specificity         ppv         npv
##   0.7647059   0.8000000   0.8125000   0.7500000

coords(mroc, .95, "threshold", ret=c("sensitivity", "specificity", "ppv", "npv"))

## sensitivity specificity         ppv         npv
##   0.7647059   0.8000000   0.8125000   0.7500000

#############################################################################################
##################

#*****************************************************************************************
************************************************

#############################################################################################
#                              effects of all variables to predict CT scan 2.5 values
#                                          a Regression analysis
#############################################################################################
#We use a linear regression analysis
#######################################
#CT.2.5 versus Scan
fix1 <- lm(CT.2.5~., data=goat_IM[,7:11])
summary(fix1)

##
## Call:
## lm(formula = CT.2.5 ~ ., data = goat_IM[, 7:11])
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -368.21  -74.49   25.18   83.34  429.97
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -333.66948  267.48321  -1.247    0.223
## scraping1         91.15355   68.17824   1.337    0.192
## Spacer.texture1   69.55075   68.89376   1.010    0.322
## CT.5               0.49610    0.03065  16.185 2.02e-15 ***
## Xray.Rank         -1.35748   10.16100  -0.134    0.895
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 185.6 on 27 degrees of freedom
## Multiple R-squared:  0.9839, Adjusted R-squared:  0.9816
## F-statistic: 413.4 on 4 and 27 DF,  p-value: < 2.2e-16
```

```
#CT.2.5 versus IM Biology

fix2 <- lm(CT.2.5~., data=goat_IM[,c(9,12:21)])
summary(fix2)
```

```
##
## Call:
## lm(formula = CT.2.5 ~ ., data = goat_IM[, c(9, 12:21)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1633.4  -688.4  -143.0   570.8  2701.5
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -1.500e+03  6.382e+03  -0.235  0.81645
## Tib_Histo_Bone.Area..3.5          3.061e+00  9.876e-01   3.100  0.00542
## IM__Histo_Vascularity            -4.372e+02  5.507e+02  -0.794  0.43615
## IM_Histo_Fibrosis                 2.485e+02  1.621e+03   0.153  0.87965
## IM_Histo_inflammatory_Cellularity -8.378e+02  6.273e+02  -1.335  0.19604
## IM_Histo_Thickness                1.661e+02  1.222e+02   1.360  0.18839
## IM.membrane._.weight              2.459e+03  3.466e+03   0.709  0.48594
## IM_Inner.Mean.cellularity         1.931e+01  1.757e+01   1.099  0.28413
## Expanded.IM_inner.cells           2.052e-02  6.700e-02   0.306  0.76242
## IM_Outer.Mean.cellularity        -9.359e-01  4.689e+01  -0.020  0.98426
## Expanded.IM_outer.cells           2.404e-02  8.502e-02   0.283  0.78013
##
## (Intercept)
## Tib_Histo_Bone.Area..3.5          **
## IM__Histo_Vascularity
## IM_Histo_Fibrosis
## IM_Histo_inflammatory_Cellularity
## IM_Histo_Thickness
## IM.membrane._.weight
```

```
## IM_Inner.Mean.cellularity
## Expanded.IM_inner.cells
## IM_Outer.Mean.cellularity
## Expanded.IM_outer.cells
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1182 on 21 degrees of freedom
## Multiple R-squared:  0.4928, Adjusted R-squared:  0.2513
## F-statistic: 2.041 on 10 and 21 DF,  p-value: 0.08101
```

```
#CT.2.5  versus gene expressions inner
fix3 <- lm(CT.2.5~., data=goat_IM[,c(9,22:40)])
summary(fix3)
```

```
##
## Call:
## lm(formula = CT.2.5 ~ ., data = goat_IM[, c(9, 22:40)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1719.3  -577.2    -2.8   417.3  2107.8
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.246e+03  1.405e+03   0.887    0.392
## IM_inner_RunX2  2.799e+00  9.155e+00   0.306    0.765
## inner_Osterix   2.095e+01  1.546e+02   0.136    0.894
## inner_PPARG    -3.387e+01  6.269e+01  -0.540    0.599
## inner_OCT4      9.869e+01  8.179e+01   1.207    0.251
## inner_Sox2      3.450e-06  4.196e-06   0.822    0.427
## inner_NANOG    -1.699e-05  8.704e-04  -0.020    0.985
## inner_ALPL      3.367e+01  9.855e+01   0.342    0.739
## inner_EGFR      2.860e+00  2.724e+00   1.050    0.315
## inner_VWF      -9.420e+01  6.069e+01  -1.552    0.147
## inner_PDGFB    -2.172e-03  4.546e-03  -0.478    0.641
## inner_TGFB1    -1.551e+00  8.287e+00  -0.187    0.855
## inner_BMP2      8.657e+01  5.832e+01   1.484    0.163
## inner_BMP6     -6.086e+01  5.994e+01  -1.015    0.330
## inner_Col1A1    5.830e-03  7.405e-03   0.787    0.446
## inner_Col2A1   -8.912e+01  1.085e+02  -0.821    0.428
## inner_Col10A1   1.919e-04  3.958e-04   0.485    0.636
## inner_GLA      -7.237e+01  1.185e+02  -0.611    0.553
## inner_IBSP     -2.091e-02  2.173e-02  -0.963    0.355
## inner_PTHR      6.808e+00  5.887e+00   1.156    0.270
##
## Residual standard error: 1333 on 12 degrees of freedom
## Multiple R-squared:  0.6317, Adjusted R-squared:  0.04859
## F-statistic: 1.083 on 19 and 12 DF,  p-value: 0.4559
```

```r
#CT.2.5  versus gene expressions outer
fix4 <- lm(CT.2.5~., data=goat_IM[,c(9,41:59)])
summary(fix4)
```

```
##
## Call:
## lm(formula = CT.2.5 ~ ., data = goat_IM[, c(9, 41:59)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1277.9  -450.0    -8.6   171.3  3445.5
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.216e+03  7.000e+02   1.736   0.1081
## outer_RunX2    1.606e+00  6.578e+00   0.244   0.8113
## outer_Osterix  4.757e+01  1.014e+02   0.469   0.6473
## outer_PPARG    1.372e+01  1.565e+01   0.877   0.3979
## outer_OCT4    -7.942e+00  6.076e+01  -0.131   0.8982
## outer_Sox2     9.720e-03  5.209e-03   1.866   0.0866 .
## outer_NANOG    1.826e-02  3.742e-02   0.488   0.6344
## outer_ALPL    -1.452e+01  1.477e+01  -0.983   0.3448
## outer_EGFR    -4.768e+01  6.241e+01  -0.764   0.4597
## outer_VWF      2.570e+00  2.174e+00   1.182   0.2601
## outer_PDGFB   -4.233e-01  8.044e-01  -0.526   0.6083
## outer_TGFB1   -8.225e+00  8.104e+00  -1.015   0.3301
## outer_BMP2     1.065e+02  4.230e+01   2.517   0.0271 *
## outer_BMP6    -9.537e+01  4.490e+01  -2.124   0.0551 .
## outer_Col1A1   6.933e-03  2.921e-03   2.373   0.0352 *
## outer_Col2A1   1.139e-01  1.150e-01   0.990   0.3415
## outer_Col10A1  4.981e-04  2.694e-04   1.849   0.0892 .
## outer_GLA     -4.633e-06  2.502e-06  -1.851   0.0889 .
## outer_IBSP     2.228e+00  6.788e+00   0.328   0.7483
## outer_PTHR     2.225e+00  4.405e+01   0.051   0.9605
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1372 on 12 degrees of freedom
## Multiple R-squared:  0.6096, Adjusted R-squared:  -0.008466
## F-statistic: 0.9863 on 19 and 12 DF,  p-value: 0.5262
```

```r
# CT.2.5 Graft, cellularity and other variables

#CT.2.5  versus gene expressions inner
fix5 <- lm(CT.2.5~., data=goat_IM[,c(9,60:70)])
summary(fix5)
```

```
##
## Call:
## lm(formula = CT.2.5 ~ ., data = goat_IM[, c(9, 60:70)])
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1638.3  -720.9  -208.6   686.4  2352.6
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -184.0733  3206.4853  -0.057   0.9548
## BMA.cellularity                    -7.8688     7.0946  -1.109   0.2805
## BMA.CTP.Prevalence                 22.6537    11.8363   1.914   0.0700 .
## MS.cellularity                      5.7240     5.2134   1.098   0.2853
## MS.CTP.Prevalence                 -11.0378     7.3159  -1.509   0.1470
## TS.cellularity                      4.0561     6.7479   0.601   0.5545
## TS.CTP.Prevalence                   2.7947     0.8762   3.190   0.0046 **
## Graft_Histo_cartilage               3.9709    21.7006   0.183   0.8567
## Graft_Histo_.bone                  45.5920    47.0641   0.969   0.3443
## Graft_Histo_hematopoetic.marrow   -13.1323    25.4095  -0.517   0.6109
## Graft_Histo_adipose.tissue      -3456.7744  1905.7676  -1.814   0.0847 .
## Graft_Histo_vasculature.area     -241.6580  3780.1364  -0.064   0.9497
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1287 on 20 degrees of freedom
## Multiple R-squared:  0.4275, Adjusted R-squared:  0.1126
## F-statistic: 1.357 on 11 and 20 DF,  p-value: 0.2659
```

#The significant varibales have a p < 0.05.

```
##############################################################################
#                                  Effects of Scraping on all variables.
##############################################################################
#We perform logistic regression
##############################################################################
# Does scraping improve CT or X-ray

fix6 <- glm(scraping~CT.2.5+CT.5+Xray.Rank,data=goat_IM, family="binomial")
summary(fix6)

##
## Call:
## glm(formula = scraping ~ CT.2.5 + CT.5 + Xray.Rank, family = "binomial",
##     data = goat_IM)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7531  -1.0428   0.4018   1.0715   1.5700
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.355017   2.843579   0.125    0.901
## CT.2.5       0.003694   0.002591   1.426    0.154
## CT.5        -0.001592   0.001207  -1.319    0.187
## Xray.Rank    0.001419   0.116538   0.012    0.990
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 44.236  on 31  degrees of freedom
## Residual deviance: 39.553  on 28  degrees of freedom
## AIC: 47.553
##
## Number of Fisher Scoring iterations: 4
```

```
#Answer: Scaping does not have any effect on CT scan or Xray
################################################################
```

```
# Does spacer TEXTURE affect CT scan or X-ray

fix7 <- glm(Spacer.texture~CT.2.5+CT.5+Xray.Rank,data=goat_IM, family="binomial")
summary(fix7)
```

```
##
## Call:
## glm(formula = Spacer.texture ~ CT.2.5 + CT.5 + Xray.Rank, family = "binomial",
##     data = goat_IM)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6744  -0.9696  -0.4603   1.1231   1.4302
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.576955   3.270741   1.705   0.0882 .
## CT.2.5       0.002589   0.002392   1.082   0.2792
## CT.5        -0.001897   0.001342  -1.413   0.1576
## Xray.Rank   -0.197213   0.122753  -1.607   0.1081
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 44.236  on 31  degrees of freedom
## Residual deviance: 39.958  on 28  degrees of freedom
## AIC: 47.958
##
## Number of Fisher Scoring iterations: 4
```

```
#Answer: SPACER_TEXTURE does not have any effect on CT scan or Xray ranking
#############################################################################
```

```
#Does spacer texture affect scraping

fix8 <- glm(Spacer.texture~scraping,data=goat_IM, family="binomial")
summary(fix8)
```

```
##
## Call:
## glm(formula = Spacer.texture ~ scraping, family = "binomial",
##     data = goat_IM)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.128  -1.128  -1.121   1.228   1.235
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.13353    0.51755  -0.258    0.796
## scraping1    0.01575    0.70991   0.022    0.982
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 44.236  on 31  degrees of freedom
## Residual deviance: 44.236  on 30  degrees of freedom
## AIC: 48.236
##
## Number of Fisher Scoring iterations: 3

#Answer: There is no significant effect of scraping on spacer texture
################################################################################

#Does spacer Texture affect IM Biology?

fix9 <- glm(Spacer.texture~.,data=goat_IM[,c(8,12:21)], family="binomial")
summary(fix9)

##
## Call:
## glm(formula = Spacer.texture ~ ., family = "binomial", data = goat_IM[,
##     c(8, 12:21)])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.64087  -0.18086  -0.00120   0.05698   1.68737
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       3.454e+01  1.582e+04   0.002    0.998
## Tib_Histo_Bone.Area..3.5          3.270e-03  4.281e-03   0.764    0.445
## IM__Histo_Vascularity             7.206e+00  4.849e+00   1.486    0.137
## IM_Histo_Fibrosis                -1.103e+01  3.956e+03  -0.003    0.998
## IM_Histo_inflammatory_Cellularity -1.607e+00  5.171e+00  -0.311    0.756
## IM_Histo_Thickness               -2.023e+00  1.411e+00  -1.434    0.152
## IM.membrane._.weight              3.308e+01  3.032e+01   1.091    0.275
## IM_Inner.Mean.cellularity        -8.494e-02  1.086e-01  -0.782    0.434
## Expanded.IM_inner.cells           2.904e-04  2.978e-04   0.975    0.330
## IM_Outer.Mean.cellularity         2.105e-01  4.541e-01   0.463    0.643
```

```
## Expanded.IM_outer.cells            7.224e-04  8.877e-04   0.814    0.416
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 44.236  on 31  degrees of freedom
## Residual deviance: 12.583  on 21  degrees of freedom
## AIC: 34.583
##
## Number of Fisher Scoring iterations: 16

#Answer: NO. Select the genes with asterix in front of the p value (Pr(>|t|))
##############################################################################

#Does spacer texture induces genes expression in inner section?

fix10 <- glm(Spacer.texture~.,data=goat_IM[,c(8,23:40)], family="binomial")

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(fix10)

##
## Call:
## glm(formula = Spacer.texture ~ ., family = "binomial", data = goat_IM[,
##     c(8, 23:40)])
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -5.533e-06  -2.323e-06  -5.987e-07   2.002e-06   5.892e-06
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.979e+01  5.158e+05       0        1
## inner_Osterix -2.015e-03  2.985e+04       0        1
## inner_PPARG   -2.450e-01  2.310e+04       0        1
## inner_OCT4     2.556e-01  4.459e+04       0        1
## inner_Sox2     8.200e-09  1.173e-03       0        1
## inner_NANOG   -6.905e-07  2.310e-01       0        1
## inner_ALPL     1.098e+00  3.266e+04       0        1
## inner_EGFR    -2.161e-02  7.732e+02       0        1
## inner_VWF     -1.060e-01  2.745e+04       0        1
## inner_PDGFB    4.761e-06  1.613e+00       0        1
## inner_TGFB1    1.730e-01  1.961e+03       0        1
## inner_BMP2    -2.663e-01  2.052e+04       0        1
## inner_BMP6     7.285e-01  3.936e+04       0        1
## inner_Col1A1   1.846e-05  1.953e+00       0        1
## inner_Col2A1  -7.139e-02  4.333e+04       0        1
## inner_Col10A1 -1.347e-06  1.345e-01       0        1
## inner_GLA      1.795e-01  3.360e+04       0        1
## inner_IBSP     1.210e-04  7.529e+00       0        1
## inner_PTHR    -4.009e-02  2.266e+03       0        1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4.4236e+01  on 31  degrees of freedom
## Residual deviance: 2.2833e-10  on 13  degrees of freedom
## AIC: 38
##
## Number of Fisher Scoring iterations: 25
```

```
#Does spacer texture induces genes expression in outer section?

fix11 <- glm(Spacer.texture~.,data=goat_IM[,c(8,41:59)], family="binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(fix11)
```

```
##
## Call:
## glm(formula = Spacer.texture ~ ., family = "binomial", data = goat_IM[,
##     c(8, 41:59)])
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -1.027e-05  -2.360e-06  -2.110e-08   1.968e-06   7.659e-06
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.783e+01  2.618e+05       0        1
## outer_RunX2    1.594e-01  2.871e+03       0        1
## outer_Osterix -9.656e-02  1.131e+05       0        1
## outer_PPARG   -2.487e-01  5.668e+03       0        1
## outer_OCT4     1.095e+00  2.559e+04       0        1
## outer_Sox2    -8.724e-05  3.766e+00       0        1
## outer_NANOG   -1.151e-03  1.660e+01       0        1
## outer_ALPL    -1.974e-01  1.074e+04       0        1
## outer_EGFR    -2.418e+00  2.943e+04       0        1
## outer_VWF     -2.384e-02  6.303e+02       0        1
## outer_PDGFB    5.173e-03  2.260e+02       0        1
## outer_TGFB1    7.689e-02  2.701e+03       0        1
## outer_BMP2     4.644e-01  4.300e+04       0        1
## outer_BMP6     4.575e-01  6.051e+04       0        1
## outer_Col1A1  -5.143e-05  3.715e+00       0        1
## outer_Col2A1   7.287e-04  3.059e+01       0        1
## outer_Col10A1 -2.722e-06  1.865e-01       0        1
## outer_GLA      4.009e-08  2.572e-03       0        1
## outer_IBSP    -1.428e-01  7.385e+03       0        1
## outer_PTHR     1.199e+00  2.732e+04       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4.4236e+01  on 31  degrees of freedom
## Residual deviance: 3.1779e-10  on 12  degrees of freedom
```

```
## AIC: 40
##
## Number of Fisher Scoring iterations: 25

#Answer: The logistic regression does not perform with the quality of the data.
# Warning: Glm fit: fitted probabilities of 0 and 1 occured. We need to find a way around
this  warning

#############################################################################################
###
#                                          INNER SECTION
#############################################################################################
##
#  DATA PREPROCESSING
### preprocess numerical features

preObj <- preProcess(goat_IM[,c(8,23:40)], method=c("scale","center"))

# transform the dataset using the parameters
transformed <- predict(preObj, goat_IM[,c(8,23:40)])
#############################################################################

#####################################################################
# Notice: the transformation did not correct for the logistic regression warning. We will
run random forest regression
#####################################################################

#############################################################################################
#Preform a randomForest regression on the transformed data

spacer.rf <- randomForest(Spacer.texture ~ ., data=transformed, mtry=3,ntree=101, proxim
ity=TRUE, oob.prox=FALSE,
                          importance=TRUE, na.action=na.omit)
 print(spacer.rf)

##
## Call:
##  randomForest(formula = Spacer.texture ~ ., data = transformed,      mtry = 3,
ntree = 101, proximity = TRUE, oob.prox = FALSE,      importance = TRUE, na.actio
n = na.omit)
##                Type of random forest: classification
##                      Number of trees: 101
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 3.12%
## Confusion matrix:
##    0  1 class.error
## 0 16  1  0.05882353
## 1  0 15  0.00000000

 predict(spacer.rf, transformed[,-1])
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1
## 26 27 28 29 30 31 32
##  1  1  1  1  1  1  1
## Levels: 0 1

# Notice: The  regression did work, and the model made a good prediction
######################################################################

# Run few more algorithms

#Using the caret package, select the most importane genes affecting spacer texture

# Boruta package
 Boruta(Spacer.texture~.,data=transformed,doTrace=2)->Bor.imbio

##  1. run of importance source...

##  2. run of importance source...

##  3. run of importance source...

##  4. run of importance source...

##  5. run of importance source...

##  6. run of importance source...

##  7. run of importance source...

##  8. run of importance source...

##  9. run of importance source...

##  10. run of importance source...

##  11. run of importance source...

## Confirmed 11 attributes: inner_ALPL, inner_BMP2, inner_BMP6, inner_Col1A1, inn
er_IBSP and 6 more.

## Rejected 1 attributes: inner_Col10A1.

##  12. run of importance source...

##  13. run of importance source...

##  14. run of importance source...

##  15. run of importance source...

## Rejected 2 attributes: inner_Col2A1, inner_EGFR.

##  16. run of importance source...

##  17. run of importance source...
```

```
##  18. run of importance source...
##  19. run of importance source...
##  20. run of importance source...
##  21. run of importance source...
##  22. run of importance source...
##  23. run of importance source...
##  24. run of importance source...
##  25. run of importance source...
## Rejected 1 attributes: inner_PPARG.
##  26. run of importance source...
##  27. run of importance source...
##  28. run of importance source...
## Rejected 1 attributes: inner_Sox2.
##  29. run of importance source...
##  30. run of importance source...
##  31. run of importance source...
##  32. run of importance source...
##  33. run of importance source...
##  34. run of importance source...
##  35. run of importance source...
##  36. run of importance source...
##  37. run of importance source...
##  38. run of importance source...
##  39. run of importance source...
##  40. run of importance source...
##  41. run of importance source...
##  42. run of importance source...
##  43. run of importance source...
##  44. run of importance source...
##  45. run of importance source...
```

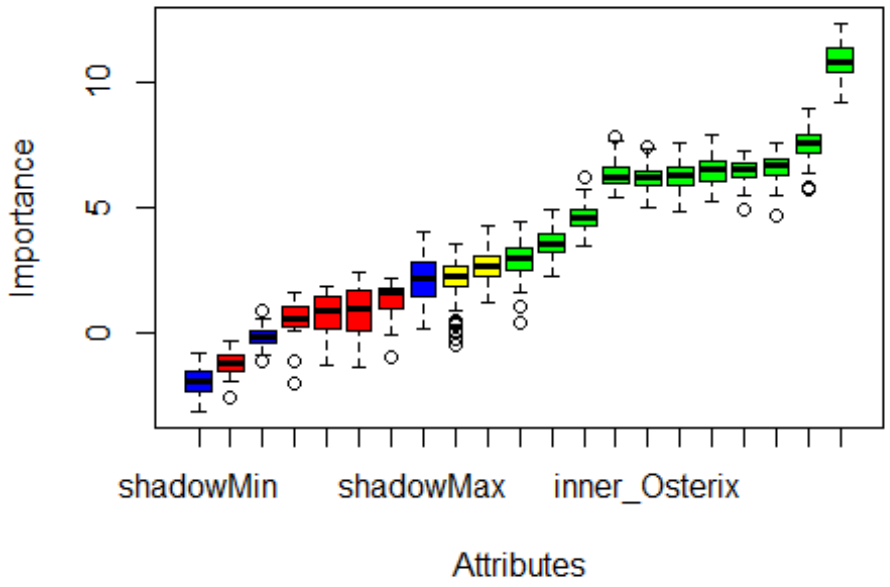
```
##  46. run of importance source...
##  47. run of importance source...
##  48. run of importance source...
##  49. run of importance source...
##  50. run of importance source...
##  51. run of importance source...
##  52. run of importance source...
##  53. run of importance source...
##  54. run of importance source...
##  55. run of importance source...
##  56. run of importance source...
##  57. run of importance source...
##  58. run of importance source...
##  59. run of importance source...
##  60. run of importance source...
##  61. run of importance source...
##  62. run of importance source...
##  63. run of importance source...
##  64. run of importance source...
##  65. run of importance source...
##  66. run of importance source...
##  67. run of importance source...
##  68. run of importance source...
##  69. run of importance source...
##  70. run of importance source...
##  71. run of importance source...
##  72. run of importance source...
##  73. run of importance source...
##  74. run of importance source...
##  75. run of importance source...
```

```
##  76. run of importance source...
##  77. run of importance source...
##  78. run of importance source...
##  79. run of importance source...
##  80. run of importance source...
##  81. run of importance source...
##  82. run of importance source...
##  83. run of importance source...
##  84. run of importance source...
##  85. run of importance source...
##  86. run of importance source...
##  87. run of importance source...
##  88. run of importance source...
##  89. run of importance source...
##  90. run of importance source...
##  91. run of importance source...
##  92. run of importance source...
##  93. run of importance source...
##  94. run of importance source...
##  95. run of importance source...
##  96. run of importance source...
##  97. run of importance source...
##  98. run of importance source...
##  99. run of importance source...
```

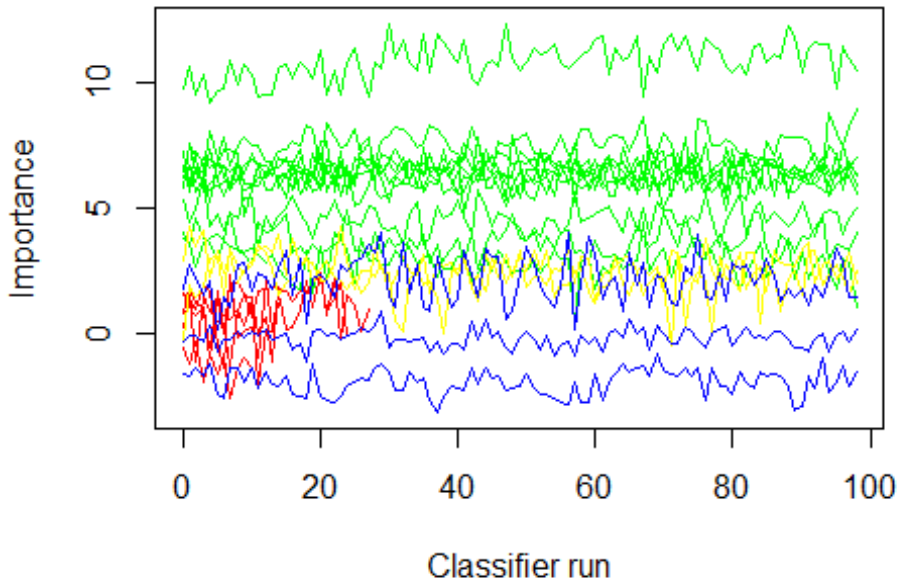
```
print(Bor.imbio,zero.print=".")
```

```
## Boruta performed 99 iterations in 2.356758 secs.
##  11 attributes confirmed important: inner_ALPL, inner_BMP2,
## inner_BMP6, inner_Col1A1, inner_IBSP and 6 more.
##  5 attributes confirmed unimportant: inner_Col10A1, inner_Col2A1,
## inner_EGFR, inner_PPARG, inner_Sox2.
##  2 tentative attributes left: inner_GLA, inner_PTHR.
```

```
plot(Bor.imbio)
```




```
plotImpHistory(Bor.imbio)
```

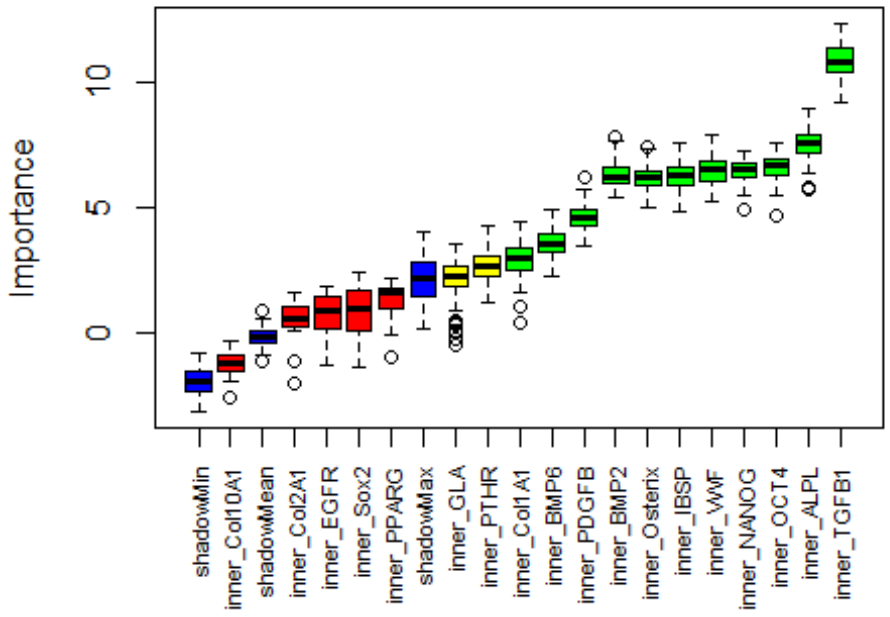



```
attStats(Bor.imbio)
```

```
##                  meanImp  medianImp     minImp     maxImp   normHits
## inner_Osterix  6.1812315  6.2164012  5.0110983  7.4308981 1.00000000
## inner_PPARG    1.2924188  1.5999425 -1.0010015  2.1764131 0.04040404
## inner_OCT4     6.6291316  6.6533770  4.6717546  7.5672878 1.00000000
```

```
## inner_Sox2     0.8364235  0.9309267 -1.4154402  2.3973797 0.05050505
## inner_NANOG    6.4785462  6.5480195  4.9093791  7.2785138 1.00000000
## inner_ALPL     7.5038228  7.5574521  5.7020341  8.9759208 1.00000000
## inner_EGFR     0.6480965  0.8537053 -1.3280557  1.8526946 0.01010101
## inner_VWF      6.4700183  6.5084605  5.2160473  7.9218791 1.00000000
## inner_PDGFB    4.5989539  4.5898938  3.4057269  6.1879881 0.98989899
## inner_TGFB1   10.8398251 10.8171807  9.2039940 12.3551338 1.00000000
## inner_BMP2     6.3149872  6.2147715  5.3510014  7.8277211 1.00000000
## inner_BMP6     3.6034479  3.4971825  2.2508265  4.9039946 0.92929293
## inner_Col1A1   2.8810938  2.9225055  0.3973978  4.4023868 0.74747475
## inner_Col2A1   0.4352938  0.5440946 -2.0416578  1.5580601 0.01010101
## inner_Col10A1 -1.2924461 -1.2471488 -2.6339468 -0.3932393 0.00000000
## inner_GLA      2.1520121  2.2389200 -0.5607250  3.5138234 0.52525253
## inner_IBSP     6.2406221  6.2524007  4.8589406  7.5883396 1.00000000
## inner_PTHR     2.6200320  2.6103547  1.1990130  4.2623316 0.63636364
##                decision
## inner_Osterix Confirmed
## inner_PPARG    Rejected
## inner_OCT4    Confirmed
## inner_Sox2     Rejected
## inner_NANOG   Confirmed
## inner_ALPL    Confirmed
## inner_EGFR     Rejected
## inner_VWF     Confirmed
## inner_PDGFB   Confirmed
## inner_TGFB1   Confirmed
## inner_BMP2    Confirmed
## inner_BMP6    Confirmed
## inner_Col1A1  Confirmed
## inner_Col2A1   Rejected
## inner_Col10A1  Rejected
## inner_GLA     Tentative
## inner_IBSP    Confirmed
## inner_PTHR    Tentative
```

```r
plot(Bor.imbio, xlab = "", xaxt = "n")
lz<-lapply(1:ncol(Bor.imbio$ImpHistory),function(i)
  Bor.imbio$ImpHistory[is.finite(Bor.imbio$ImpHistory[,i]),i])
names(lz) <- colnames(Bor.imbio$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(side = 1,las=2,labels = names(Labels),
     at = 1:ncol(Bor.imbio$ImpHistory), cex.axis = 0.7)
```




```
#Random Forest
modelFita <- train(Spacer.texture~.,data=transformed, method="rf" )  # random forest:  ge
nes are ranked from most important to least
varImp(modelFita)
```

```
## rf variable importance
##
##                Overall
## inner_TGFB1   100.000
## inner_Osterix  82.458
## inner_BMP2     78.165
## inner_IBSP     70.208
## inner_VWF      67.711
## inner_ALPL     67.053
## inner_OCT4     56.375
## inner_PDGFB    50.397
## inner_NANOG    47.784
## inner_GLA      22.651
## inner_PTHR     20.630
## inner_BMP6     17.523
## inner_Col1A1   16.480
## inner_Col2A1   15.881
## inner_PPARG     2.805
## inner_Sox2      2.324
## inner_EGFR      0.465
## inner_Col10A1   0.000
```

```
######################
modelFitb <- train(Spacer.texture~.,data=transformed, method="glmnet" )# lasso
```

```
## Loading required package: glmnet

## Loading required package: Matrix

## Loaded glmnet 2.0-5

##
## Attaching package: 'glmnet'

## The following object is masked from 'package:pROC':
##
##     auc
```

```
varImp(modelFitb)
```

```
## glmnet variable importance
##
##                Overall
## inner_TGFB1   100.0000
## inner_ALPL     15.4794
## inner_PDGFB     0.5116
## inner_PPARG     0.0000
## inner_Col10A1   0.0000
## inner_GLA       0.0000
## inner_Sox2      0.0000
## inner_PTHR      0.0000
## inner_EGFR      0.0000
## inner_IBSP      0.0000
## inner_Col2A1    0.0000
## inner_NANOG     0.0000
## inner_VWF       0.0000
## inner_Osterix   0.0000
## inner_Col1A1    0.0000
## inner_BMP2      0.0000
## inner_BMP6      0.0000
## inner_OCT4      0.0000
```

```
#####################################################################
modelFitc <- train(Spacer.texture~.,data=transformed, method="pda" )  #penalyzed discrimi
nant analysis
```

```
## Loading required package: mda

## Warning: package 'mda' was built under R version 3.3.2

## Loading required package: class

## Loaded mda 0.4-9

## Warning: predictions failed for Resample04: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments
```

```
## Warning: predictions failed for Resample07: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```

```
varImp(modelFitc)
```

```
## ROC curve variable importance
##
##                Importance
## inner_TGFB1        100.00
## inner_ALPL          98.92
## inner_OCT4          96.76
## inner_VWF           95.14
## inner_NANOG         94.59
## inner_BMP2          92.97
## inner_Osterix       92.43
## inner_IBSP          92.43
## inner_PDGFB         91.35
## inner_BMP6          87.03
## inner_Col1A1        86.49
## inner_PTHR          84.32
## inner_GLA           83.24
## inner_Col2A1        82.70
## inner_EGFR          76.49
## inner_PPARG         74.59
## inner_Col10A1       41.62
## inner_Sox2           0.00
```

```
#####################################################################
modelFitd <- train(Spacer.texture~.,data=transformed, method="nnet" )  #Neural Network
```

```
## # weights:  21
## initial  value 21.726105
## iter  10 value 0.000170
## iter  10 value 0.000086
## iter  10 value 0.000086
## final  value 0.000086
## converged
## # weights:  61
## initial  value 22.307856
## iter  10 value 0.018399
## iter  20 value 0.006195
## final  value 0.000099
## converged
## # weights:  101
## initial  value 19.347663
## iter  10 value 0.000745
## final  value 0.000093
## converged
```

```
## # weights:  21
## initial  value 22.808778
## iter  10 value 6.357630
## iter  20 value 5.993620
## final  value 5.993620
## converged
## # weights:  61
## initial  value 19.670308
## iter  10 value 3.977614
## iter  20 value 3.838529
## final  value 3.838502
## converged
## # weights:  101
## initial  value 24.168160
## iter  10 value 3.312420
## iter  20 value 3.237484
## final  value 3.237402
## converged
## # weights:  21
## initial  value 19.891085
## iter  10 value 0.527806
## iter  20 value 0.055420
## iter  30 value 0.048312
## iter  40 value 0.044042
## iter  50 value 0.042976
## iter  60 value 0.042770
## iter  70 value 0.042724
## iter  80 value 0.042703
## iter  90 value 0.042698
## iter 100 value 0.042695
## final  value 0.042695
## stopped after 100 iterations
## # weights:  61
## initial  value 16.534028
## iter  10 value 0.079981
## iter  20 value 0.030115
## iter  30 value 0.026671
## iter  40 value 0.020609
## iter  50 value 0.019785
## iter  60 value 0.018739
## iter  70 value 0.017631
## iter  80 value 0.017563
## iter  90 value 0.017397
## iter 100 value 0.017382
## final  value 0.017382
## stopped after 100 iterations
## # weights:  101
## initial  value 22.958267
## iter  10 value 0.038122
## iter  20 value 0.027438
```

```
## iter  30 value 0.019591
## iter  40 value 0.016464
## iter  50 value 0.015644
## iter  60 value 0.015318
## iter  70 value 0.014238
## iter  80 value 0.014011
## iter  90 value 0.013157
## iter 100 value 0.012802
## final  value 0.012802
## stopped after 100 iterations
## # weights:  21
## initial  value 21.845568
## iter  10 value 4.004855
## iter  20 value 3.803403
## final  value 3.803208
## converged
## # weights:  61
## initial  value 25.355068
## iter  10 value 3.937335
## iter  20 value 1.286308
## iter  30 value 0.002072
## final  value 0.000065
## converged
## # weights:  101
## initial  value 23.768550
## iter  10 value 0.037832
## final  value 0.000093
## converged
## # weights:  21
## initial  value 23.809201
## iter  10 value 8.595224
## iter  20 value 6.283591
## final  value 6.283587
## converged
## # weights:  61
## initial  value 23.415868
## iter  10 value 5.519295
## iter  20 value 4.715056
## final  value 4.714914
## converged
## # weights:  101
## initial  value 27.152378
## iter  10 value 4.299456
## iter  20 value 3.173214
## iter  30 value 3.169619
## final  value 3.169614
## converged
## # weights:  21
## initial  value 22.113667
## iter  10 value 0.045074
```

```
## iter  20 value 0.044240
## iter  30 value 0.044060
## iter  40 value 0.043893
## iter  50 value 0.043744
## iter  60 value 0.043701
## iter  70 value 0.043680
## iter  80 value 0.043679
## final  value 0.043678
## converged
## # weights:  61
## initial  value 23.757153
## iter  10 value 0.963554
## iter  20 value 0.072809
## iter  30 value 0.050771
## iter  40 value 0.037455
## iter  50 value 0.036666
## iter  60 value 0.036402
## iter  70 value 0.030585
## iter  80 value 0.027149
## iter  90 value 0.024079
## iter 100 value 0.021640
## final  value 0.021640
## stopped after 100 iterations
## # weights:  101
## initial  value 19.465720
## iter  10 value 0.044990
## iter  20 value 0.038388
## iter  30 value 0.031393
## iter  40 value 0.019557
## iter  50 value 0.017244
## iter  60 value 0.014802
## iter  70 value 0.014081
## iter  80 value 0.013299
## iter  90 value 0.012719
## iter 100 value 0.012539
## final  value 0.012539
## stopped after 100 iterations
## # weights:  21
## initial  value 20.439694
## iter  10 value 0.004001
## iter  20 value 0.000306
## final  value 0.000085
## converged
## # weights:  61
## initial  value 19.714875
## iter  10 value 0.011563
## final  value 0.000069
## converged
## # weights:  101
## initial  value 26.177510
```

```
## iter  10 value 0.003101
## final  value 0.000066
## converged
## # weights:  21
## initial  value 24.908882
## iter  10 value 6.287174
## iter  20 value 6.210582
## iter  20 value 6.210582
## iter  20 value 6.210582
## final  value 6.210582
## converged
## # weights:  61
## initial  value 19.462959
## iter  10 value 4.124074
## iter  20 value 3.610056
## final  value 3.610017
## converged
## # weights:  101
## initial  value 28.278064
## iter  10 value 3.507264
## iter  20 value 2.975263
## iter  30 value 2.974283
## final  value 2.974283
## converged
## # weights:  21
## initial  value 19.853270
## iter  10 value 0.459168
## iter  20 value 0.062066
## iter  30 value 0.055739
## iter  40 value 0.044650
## iter  50 value 0.044414
## iter  60 value 0.043939
## iter  70 value 0.043781
## iter  80 value 0.043553
## iter  90 value 0.043537
## iter 100 value 0.043536
## final  value 0.043536
## stopped after 100 iterations
## # weights:  61
## initial  value 19.126367
## iter  10 value 0.046936
## iter  20 value 0.038125
## iter  30 value 0.033721
## iter  40 value 0.030122
## iter  50 value 0.029560
## iter  60 value 0.029169
## iter  70 value 0.028230
## iter  80 value 0.026460
## iter  90 value 0.025848
## iter 100 value 0.024782
```

```
## final  value 0.024782
## stopped after 100 iterations
## # weights:  101
## initial  value 21.192208
## iter  10 value 0.089565
## iter  20 value 0.045774
## iter  30 value 0.028897
## iter  40 value 0.024204
## iter  50 value 0.023372
## iter  60 value 0.018760
## iter  70 value 0.017659
## iter  80 value 0.017219
## iter  90 value 0.016259
## iter 100 value 0.014602
## final  value 0.014602
## stopped after 100 iterations
## # weights:  21
## initial  value 24.667697
## iter  10 value 0.004617
## final  value 0.000074
## converged
## # weights:  61
## initial  value 22.934692
## iter  10 value 0.001719
## final  value 0.000072
## converged
## # weights:  101
## initial  value 24.746311
## iter  10 value 0.001386
## iter  20 value 0.000143
## final  value 0.000082
## converged
## # weights:  21
## initial  value 20.926776
## iter  10 value 6.305834
## final  value 6.302860
## converged
## # weights:  61
## initial  value 22.394105
## iter  10 value 3.726397
## final  value 3.721374
## converged
## # weights:  101
## initial  value 25.803178
## iter  10 value 3.197353
## iter  20 value 3.138050
## final  value 3.137931
## converged
## # weights:  21
## initial  value 24.335873
```

```
## iter  10 value 0.047027
## iter  20 value 0.045738
## iter  30 value 0.044659
## iter  40 value 0.044011
## iter  50 value 0.043934
## iter  60 value 0.043840
## iter  70 value 0.043805
## iter  80 value 0.043797
## final  value 0.043796
## converged
## # weights:  61
## initial  value 26.195889
## iter  10 value 0.026026
## iter  20 value 0.021382
## iter  30 value 0.019883
## iter  40 value 0.018334
## iter  50 value 0.017693
## iter  60 value 0.017443
## iter  70 value 0.017342
## iter  80 value 0.017309
## iter  90 value 0.017291
## iter 100 value 0.017290
## final  value 0.017290
## stopped after 100 iterations
## # weights:  101
## initial  value 25.997669
## iter  10 value 0.025801
## iter  20 value 0.020773
## iter  30 value 0.016071
## iter  40 value 0.014377
## iter  50 value 0.013615
## iter  60 value 0.012755
## iter  70 value 0.012577
## iter  80 value 0.012469
## iter  90 value 0.012454
## iter 100 value 0.012449
## final  value 0.012449
## stopped after 100 iterations
## # weights:  21
## initial  value 22.178402
## iter  10 value 3.874354
## iter  20 value 3.862077
## final  value 3.862065
## converged
## # weights:  61
## initial  value 23.097104
## iter  10 value 0.003284
## final  value 0.000071
## converged
## # weights:  101
```

```
## initial  value 21.285608
## iter  10 value 0.040962
## final  value 0.000057
## converged
## # weights:  21
## initial  value 22.482774
## iter  10 value 8.869162
## iter  20 value 6.544432
## iter  30 value 6.268096
## final  value 6.268096
## converged
## # weights:  61
## initial  value 25.475362
## iter  10 value 5.647464
## iter  20 value 3.811076
## iter  30 value 3.770992
## iter  40 value 3.766289
## final  value 3.766285
## converged
## # weights:  101
## initial  value 30.567451
## iter  10 value 3.941015
## iter  20 value 3.211724
## iter  30 value 3.207768
## final  value 3.207758
## converged
## # weights:  21
## initial  value 21.909354
## iter  10 value 3.911676
## iter  20 value 3.907837
## iter  30 value 3.897531
## iter  40 value 3.887925
## iter  50 value 3.885752
## iter  60 value 3.882973
## iter  70 value 3.882239
## iter  80 value 3.881658
## iter  90 value 0.121409
## iter 100 value 0.047107
## final  value 0.047107
## stopped after 100 iterations
## # weights:  61
## initial  value 24.017088
## iter  10 value 0.048055
## iter  20 value 0.035949
## iter  30 value 0.026039
## iter  40 value 0.024239
## iter  50 value 0.022454
## iter  60 value 0.021326
## iter  70 value 0.020859
## iter  80 value 0.019244
```

```
## iter  90 value 0.019004
## iter 100 value 0.018055
## final  value 0.018055
## stopped after 100 iterations
## # weights:  101
## initial  value 24.416374
## iter  10 value 0.189356
## iter  20 value 0.078284
## iter  30 value 0.041348
## iter  40 value 0.036652
## iter  50 value 0.034824
## iter  60 value 0.027974
## iter  70 value 0.024867
## iter  80 value 0.022364
## iter  90 value 0.019525
## iter 100 value 0.018704
## final  value 0.018704
## stopped after 100 iterations
## # weights:  21
## initial  value 21.664372
## iter  10 value 0.021521
## final  value 0.000085
## converged
## # weights:  61
## initial  value 26.049161
## iter  10 value 0.003866
## final  value 0.000075
## converged
## # weights:  101
## initial  value 28.554524
## final  value 0.000000
## converged
## # weights:  21
## initial  value 20.649311
## iter  10 value 6.130848
## iter  20 value 6.043825
## final  value 6.043825
## converged
## # weights:  61
## initial  value 25.493283
## iter  10 value 3.936651
## iter  20 value 3.672328
## final  value 3.672247
## converged
## # weights:  101
## initial  value 25.282481
## iter  10 value 3.420231
## iter  20 value 3.320865
## final  value 3.320447
## converged
```

```
## # weights:  21
## initial  value 25.798453
## iter  10 value 0.421377
## iter  20 value 0.326001
## iter  30 value 0.187289
## iter  40 value 0.136598
## iter  50 value 0.086089
## iter  60 value 0.065882
## iter  70 value 0.045254
## iter  80 value 0.043975
## iter  90 value 0.043705
## iter 100 value 0.043701
## final  value 0.043701
## stopped after 100 iterations
## # weights:  61
## initial  value 24.001645
## iter  10 value 0.027796
## iter  20 value 0.021218
## iter  30 value 0.018399
## iter  40 value 0.017703
## iter  50 value 0.017439
## iter  60 value 0.017217
## iter  70 value 0.017187
## iter  80 value 0.017168
## iter  90 value 0.017156
## iter 100 value 0.017154
## final  value 0.017154
## stopped after 100 iterations
## # weights:  101
## initial  value 31.152172
## iter  10 value 0.063414
## iter  20 value 0.034036
## iter  30 value 0.025649
## iter  40 value 0.017875
## iter  50 value 0.015494
## iter  60 value 0.014530
## iter  70 value 0.013098
## iter  80 value 0.012616
## iter  90 value 0.012484
## iter 100 value 0.012396
## final  value 0.012396
## stopped after 100 iterations
## # weights:  21
## initial  value 21.580752
## iter  10 value 0.004011
## final  value 0.000063
## converged
## # weights:  61
## initial  value 20.814020
## iter  10 value 0.069342
```

```
## iter  20 value 0.001110
## final  value 0.000059
## converged
## # weights:  101
## initial  value 29.022104
## iter  10 value 0.005597
## iter  20 value 0.000290
## final  value 0.000098
## converged
## # weights:  21
## initial  value 25.467747
## iter  10 value 5.891105
## final  value 5.873248
## converged
## # weights:  61
## initial  value 19.575862
## iter  10 value 3.510918
## iter  20 value 3.485690
## final  value 3.485688
## converged
## # weights:  101
## initial  value 24.143926
## iter  10 value 3.178467
## iter  20 value 2.930178
## final  value 2.930094
## converged
## # weights:  21
## initial  value 26.609564
## iter  10 value 5.812815
## iter  20 value 0.401612
## iter  30 value 0.049328
## iter  40 value 0.043459
## iter  50 value 0.042603
## iter  60 value 0.042485
## iter  70 value 0.042458
## iter  80 value 0.042452
## iter  90 value 0.042447
## iter 100 value 0.042446
## final  value 0.042446
## stopped after 100 iterations
## # weights:  61
## initial  value 26.847542
## iter  10 value 0.091003
## iter  20 value 0.080592
## iter  30 value 0.061460
## iter  40 value 0.043334
## iter  50 value 0.028110
## iter  60 value 0.022169
## iter  70 value 0.021665
## iter  80 value 0.019928
```

```
## iter  90 value 0.017922
## iter 100 value 0.017743
## final  value 0.017743
## stopped after 100 iterations
## # weights:  101
## initial  value 20.206778
## iter  10 value 0.115103
## iter  20 value 0.076623
## iter  30 value 0.041207
## iter  40 value 0.037670
## iter  50 value 0.029409
## iter  60 value 0.020255
## iter  70 value 0.019303
## iter  80 value 0.014897
## iter  90 value 0.014092
## iter 100 value 0.013651
## final  value 0.013651
## stopped after 100 iterations
## # weights:  21
## initial  value 22.096731
## iter  10 value 0.329501
## iter  20 value 0.000320
## final  value 0.000080
## converged
## # weights:  61
## initial  value 22.958803
## final  value 0.000018
## converged
## # weights:  101
## initial  value 26.652828
## iter  10 value 0.001646
## final  value 0.000053
## converged
## # weights:  21
## initial  value 26.084043
## iter  10 value 6.404532
## iter  20 value 6.380933
## final  value 6.380932
## converged
## # weights:  61
## initial  value 26.530708
## iter  10 value 4.764707
## iter  20 value 4.709492
## final  value 4.709475
## converged
## # weights:  101
## initial  value 28.708934
## iter  10 value 3.590854
## iter  20 value 3.244575
## iter  30 value 3.244191
```

```
## final  value 3.244191
## converged
## # weights:  21
## initial  value 23.231587
## iter  10 value 0.107868
## iter  20 value 0.071802
## iter  30 value 0.049825
## iter  40 value 0.049028
## iter  50 value 0.047237
## iter  60 value 0.043848
## iter  70 value 0.043489
## iter  80 value 0.043411
## iter  90 value 0.043399
## iter 100 value 0.043399
## final  value 0.043399
## stopped after 100 iterations
## # weights:  61
## initial  value 27.927225
## iter  10 value 0.115497
## iter  20 value 0.074534
## iter  30 value 0.057314
## iter  40 value 0.041385
## iter  50 value 0.038431
## iter  60 value 0.033941
## iter  70 value 0.030525
## iter  80 value 0.029576
## iter  90 value 0.029298
## iter 100 value 0.028151
## final  value 0.028151
## stopped after 100 iterations
## # weights:  101
## initial  value 28.649743
## iter  10 value 0.023280
## iter  20 value 0.019237
## iter  30 value 0.016698
## iter  40 value 0.015354
## iter  50 value 0.014860
## iter  60 value 0.014553
## iter  70 value 0.014478
## iter  80 value 0.014447
## iter  90 value 0.014427
## iter 100 value 0.014421
## final  value 0.014421
## stopped after 100 iterations
## # weights:  21
## initial  value 23.039776
## iter  10 value 5.085891
## final  value 3.673973
## converged
## # weights:  61
```

```
## initial  value 26.488178
## iter  10 value 0.250516
## iter  20 value 0.000894
## final  value 0.000055
## converged
## # weights:  101
## initial  value 21.087728
## iter  10 value 0.001647
## final  value 0.000079
## converged
## # weights:  21
## initial  value 25.365359
## iter  10 value 6.519910
## iter  20 value 6.240119
## final  value 6.240099
## converged
## # weights:  61
## initial  value 22.980311
## iter  10 value 5.414477
## iter  20 value 4.683678
## final  value 4.683414
## converged
## # weights:  101
## initial  value 23.503317
## iter  10 value 3.612321
## iter  20 value 3.249834
## final  value 3.249434
## converged
## # weights:  21
## initial  value 23.410233
## iter  10 value 0.060420
## iter  20 value 0.055289
## iter  30 value 0.046205
## iter  40 value 0.044956
## iter  50 value 0.044383
## iter  60 value 0.044096
## iter  70 value 0.044021
## iter  80 value 0.043957
## iter  90 value 0.043940
## iter 100 value 0.043939
## final  value 0.043939
## stopped after 100 iterations
## # weights:  61
## initial  value 22.121860
## iter  10 value 0.046811
## iter  20 value 0.034563
## iter  30 value 0.026824
## iter  40 value 0.024166
## iter  50 value 0.021253
## iter  60 value 0.019192
```

```
## iter  70 value 0.018441
## iter  80 value 0.017861
## iter  90 value 0.017643
## iter 100 value 0.017538
## final  value 0.017538
## stopped after 100 iterations
## # weights:  101
## initial  value 30.745579
## iter  10 value 0.024520
## iter  20 value 0.021872
## iter  30 value 0.016604
## iter  40 value 0.015594
## iter  50 value 0.014343
## iter  60 value 0.013629
## iter  70 value 0.012901
## iter  80 value 0.012759
## iter  90 value 0.012691
## iter 100 value 0.012663
## final  value 0.012663
## stopped after 100 iterations
## # weights:  21
## initial  value 25.696052
## iter  10 value 3.474783
## iter  20 value 0.137877
## iter  30 value 0.001036
## final  value 0.000068
## converged
## # weights:  61
## initial  value 23.358189
## iter  10 value 0.012142
## final  value 0.000067
## converged
## # weights:  101
## initial  value 28.289433
## iter  10 value 0.017033
## final  value 0.000088
## converged
## # weights:  21
## initial  value 22.608201
## iter  10 value 6.037002
## final  value 6.035248
## converged
## # weights:  61
## initial  value 25.222686
## iter  10 value 3.891372
## iter  20 value 3.706214
## iter  30 value 3.705967
## final  value 3.705967
## converged
## # weights:  101
```

```
## initial  value 24.660306
## iter  10 value 3.619805
## iter  20 value 3.184284
## iter  30 value 3.177991
## final  value 3.177990
## converged
## # weights:  21
## initial  value 20.258572
## iter  10 value 0.048874
## iter  20 value 0.046589
## iter  30 value 0.044565
## iter  40 value 0.043887
## iter  50 value 0.043628
## iter  60 value 0.043030
## iter  70 value 0.042908
## iter  80 value 0.042899
## iter  90 value 0.042896
## iter 100 value 0.042895
## final  value 0.042895
## stopped after 100 iterations
## # weights:  61
## initial  value 22.958515
## iter  10 value 3.470961
## iter  20 value 0.138758
## iter  30 value 0.054693
## iter  40 value 0.045366
## iter  50 value 0.037722
## iter  60 value 0.035541
## iter  70 value 0.030579
## iter  80 value 0.029116
## iter  90 value 0.027008
## iter 100 value 0.025437
## final  value 0.025437
## stopped after 100 iterations
## # weights:  101
## initial  value 22.806406
## iter  10 value 0.058963
## iter  20 value 0.041879
## iter  30 value 0.027174
## iter  40 value 0.020489
## iter  50 value 0.016528
## iter  60 value 0.014769
## iter  70 value 0.014423
## iter  80 value 0.014015
## iter  90 value 0.013553
## iter 100 value 0.013142
## final  value 0.013142
## stopped after 100 iterations
## # weights:  21
## initial  value 19.016461
```

```
## iter  10 value 0.007307
## final  value 0.000083
## converged
## # weights:  61
## initial  value 23.140410
## iter  10 value 0.003894
## iter  20 value 0.001002
## final  value 0.000079
## converged
## # weights:  101
## initial  value 18.978145
## final  value 0.000000
## converged
## # weights:  21
## initial  value 28.637870
## iter  10 value 6.434822
## iter  20 value 6.367969
## iter  20 value 6.367969
## iter  20 value 6.367969
## final  value 6.367969
## converged
## # weights:  61
## initial  value 22.430016
## iter  10 value 4.411136
## iter  20 value 3.804960
## iter  30 value 3.803864
## final  value 3.803863
## converged
## # weights:  101
## initial  value 26.236961
## iter  10 value 3.562828
## iter  20 value 3.293841
## iter  30 value 3.180224
## iter  40 value 3.179916
## final  value 3.179916
## converged
## # weights:  21
## initial  value 22.433388
## iter  10 value 3.602567
## iter  20 value 0.119496
## iter  30 value 0.075979
## iter  40 value 0.061481
## iter  50 value 0.059285
## iter  60 value 0.054250
## iter  70 value 0.044614
## iter  80 value 0.044284
## iter  90 value 0.044067
## iter 100 value 0.044022
## final  value 0.044022
## stopped after 100 iterations
```

```
## # weights:  61
## initial  value 25.339935
## iter  10 value 0.071046
## iter  20 value 0.057157
## iter  30 value 0.044763
## iter  40 value 0.041631
## iter  50 value 0.036241
## iter  60 value 0.027101
## iter  70 value 0.023679
## iter  80 value 0.021470
## iter  90 value 0.019469
## iter 100 value 0.019023
## final  value 0.019023
## stopped after 100 iterations
## # weights:  101
## initial  value 24.865127
## iter  10 value 0.092787
## iter  20 value 0.070835
## iter  30 value 0.059969
## iter  40 value 0.025824
## iter  50 value 0.020600
## iter  60 value 0.019483
## iter  70 value 0.016623
## iter  80 value 0.015435
## iter  90 value 0.014978
## iter 100 value 0.013961
## final  value 0.013961
## stopped after 100 iterations
## # weights:  21
## initial  value 24.575718
## iter  10 value 0.476749
## iter  20 value 0.000964
## final  value 0.000062
## converged
## # weights:  61
## initial  value 23.153230
## iter  10 value 0.067507
## iter  20 value 0.000180
## iter  20 value 0.000090
## iter  20 value 0.000090
## final  value 0.000090
## converged
## # weights:  101
## initial  value 24.539328
## iter  10 value 0.041388
## iter  20 value 0.000547
## final  value 0.000079
## converged
## # weights:  21
## initial  value 22.809678
```

```
## iter  10 value 6.608501
## iter  20 value 6.375111
## final  value 6.374635
## converged
## # weights:  61
## initial  value 27.533459
## iter  10 value 4.673624
## iter  20 value 3.872106
## iter  30 value 3.866923
## final  value 3.866923
## converged
## # weights:  101
## initial  value 21.826857
## iter  10 value 3.429817
## iter  20 value 3.311339
## final  value 3.311238
## converged
## # weights:  21
## initial  value 22.203076
## iter  10 value 3.858157
## iter  20 value 3.851852
## iter  30 value 3.837826
## iter  40 value 0.117145
## iter  50 value 0.052243
## iter  60 value 0.046406
## iter  70 value 0.044156
## iter  80 value 0.043977
## iter  90 value 0.043930
## iter 100 value 0.043908
## final  value 0.043908
## stopped after 100 iterations
## # weights:  61
## initial  value 22.002835
## iter  10 value 2.370555
## iter  20 value 0.069184
## iter  30 value 0.065113
## iter  40 value 0.042981
## iter  50 value 0.039811
## iter  60 value 0.035918
## iter  70 value 0.032984
## iter  80 value 0.029082
## iter  90 value 0.026469
## iter 100 value 0.026042
## final  value 0.026042
## stopped after 100 iterations
## # weights:  101
## initial  value 25.100128
## iter  10 value 0.044602
## iter  20 value 0.030509
## iter  30 value 0.023402
```

```
## iter  40 value 0.019402
## iter  50 value 0.017219
## iter  60 value 0.015587
## iter  70 value 0.015255
## iter  80 value 0.015175
## iter  90 value 0.014842
## iter 100 value 0.014730
## final  value 0.014730
## stopped after 100 iterations
## # weights:  21
## initial  value 22.957224
## iter  10 value 0.210964
## iter  20 value 0.000188
## iter  20 value 0.000095
## iter  20 value 0.000095
## final  value 0.000095
## converged
## # weights:  61
## initial  value 22.602836
## iter  10 value 3.967904
## iter  20 value 3.717551
## iter  30 value 0.158471
## iter  40 value 0.000247
## final  value 0.000096
## converged
## # weights:  101
## initial  value 26.535790
## iter  10 value 0.002064
## iter  20 value 0.000306
## final  value 0.000085
## converged
## # weights:  21
## initial  value 22.070045
## iter  10 value 8.534897
## iter  20 value 6.388186
## final  value 6.388182
## converged
## # weights:  61
## initial  value 22.567883
## iter  10 value 3.890069
## iter  20 value 3.839207
## final  value 3.839202
## converged
## # weights:  101
## initial  value 24.504496
## iter  10 value 3.820048
## iter  20 value 3.273819
## iter  30 value 3.261202
## iter  40 value 3.260320
## final  value 3.260319
```

```
## converged
## # weights:  21
## initial  value 22.724669
## iter  10 value 3.945363
## iter  20 value 3.943790
## iter  30 value 3.939484
## iter  40 value 0.079298
## iter  50 value 0.051512
## iter  60 value 0.047406
## iter  70 value 0.044358
## iter  80 value 0.044135
## iter  90 value 0.044015
## iter 100 value 0.043974
## final  value 0.043974
## stopped after 100 iterations
## # weights:  61
## initial  value 21.599187
## iter  10 value 1.295708
## iter  20 value 0.037439
## iter  30 value 0.031143
## iter  40 value 0.020857
## iter  50 value 0.018886
## iter  60 value 0.018501
## iter  70 value 0.017872
## iter  80 value 0.017749
## iter  90 value 0.017630
## iter 100 value 0.017594
## final  value 0.017594
## stopped after 100 iterations
## # weights:  101
## initial  value 24.463377
## iter  10 value 0.269111
## iter  20 value 0.040772
## iter  30 value 0.028060
## iter  40 value 0.021969
## iter  50 value 0.019727
## iter  60 value 0.016555
## iter  70 value 0.014860
## iter  80 value 0.013990
## iter  90 value 0.013545
## iter 100 value 0.013281
## final  value 0.013281
## stopped after 100 iterations
## # weights:  21
## initial  value 21.109154
## iter  10 value 0.037840
## final  value 0.000052
## converged
## # weights:  61
## initial  value 26.591285
```

```
## iter  10 value 0.005743
## final  value 0.000097
## converged
## # weights:  101
## initial  value 20.855315
## iter  10 value 0.000826
## final  value 0.000051
## converged
## # weights:  21
## initial  value 22.713905
## iter  10 value 6.280137
## final  value 6.280011
## converged
## # weights:  61
## initial  value 32.650982
## iter  10 value 4.121754
## iter  20 value 3.696693
## iter  30 value 3.694952
## final  value 3.694952
## converged
## # weights:  101
## initial  value 30.637190
## iter  10 value 3.079558
## iter  20 value 3.049878
## final  value 3.049872
## converged
## # weights:  21
## initial  value 19.162935
## iter  10 value 0.057533
## iter  20 value 0.045579
## iter  30 value 0.042756
## iter  40 value 0.042622
## iter  50 value 0.042504
## iter  60 value 0.042466
## iter  70 value 0.042428
## iter  80 value 0.042425
## final  value 0.042424
## converged
## # weights:  61
## initial  value 21.311799
## iter  10 value 0.025840
## iter  20 value 0.022525
## iter  30 value 0.018985
## iter  40 value 0.017968
## iter  50 value 0.017462
## iter  60 value 0.017398
## iter  70 value 0.017343
## iter  80 value 0.017323
## iter  90 value 0.017309
## iter 100 value 0.017301
```

```
## final  value 0.017301
## stopped after 100 iterations
## # weights:  101
## initial  value 21.300454
## iter  10 value 0.033209
## iter  20 value 0.026916
## iter  30 value 0.019672
## iter  40 value 0.015400
## iter  50 value 0.013451
## iter  60 value 0.012922
## iter  70 value 0.012648
## iter  80 value 0.012521
## iter  90 value 0.012462
## iter 100 value 0.012446
## final  value 0.012446
## stopped after 100 iterations
## # weights:  21
## initial  value 24.721903
## iter  10 value 3.917649
## iter  10 value 3.917649
## iter  10 value 3.917649
## final  value 3.917649
## converged
## # weights:  61
## initial  value 23.284548
## iter  10 value 0.022289
## final  value 0.000062
## converged
## # weights:  101
## initial  value 31.584917
## iter  10 value 0.009855
## final  value 0.000082
## converged
## # weights:  21
## initial  value 27.117066
## iter  10 value 8.449327
## iter  20 value 6.439684
## final  value 6.439677
## converged
## # weights:  61
## initial  value 27.766085
## iter  10 value 3.998087
## iter  20 value 3.907757
## final  value 3.907741
## converged
## # weights:  101
## initial  value 23.164880
## iter  10 value 3.613515
## iter  20 value 3.337490
## iter  30 value 3.331998
```

```
## final  value 3.331998
## converged
## # weights:  21
## initial  value 23.726706
## iter  10 value 4.072254
## iter  20 value 4.028166
## iter  30 value 3.991831
## iter  40 value 3.967656
## iter  50 value 3.960085
## iter  60 value 3.955450
## iter  70 value 3.939821
## iter  80 value 3.937881
## iter  90 value 3.937470
## iter 100 value 0.107161
## final  value 0.107161
## stopped after 100 iterations
## # weights:  61
## initial  value 20.798414
## iter  10 value 0.052930
## iter  20 value 0.043421
## iter  30 value 0.031657
## iter  40 value 0.024042
## iter  50 value 0.022043
## iter  60 value 0.019506
## iter  70 value 0.018857
## iter  80 value 0.018573
## iter  90 value 0.018392
## iter 100 value 0.017989
## final  value 0.017989
## stopped after 100 iterations
## # weights:  101
## initial  value 24.869102
## iter  10 value 0.079400
## iter  20 value 0.070166
## iter  30 value 0.037230
## iter  40 value 0.026193
## iter  50 value 0.022033
## iter  60 value 0.018461
## iter  70 value 0.016549
## iter  80 value 0.015873
## iter  90 value 0.014890
## iter 100 value 0.013674
## final  value 0.013674
## stopped after 100 iterations
## # weights:  21
## initial  value 22.558062
## iter  10 value 3.943766
## iter  20 value 0.019216
## final  value 0.000095
## converged
```

```
## # weights:  61
## initial  value 25.251418
## iter  10 value 0.145805
## iter  20 value 0.001166
## final  value 0.000059
## converged
## # weights:  101
## initial  value 28.615906
## iter  10 value 0.000174
## final  value 0.000080
## converged
## # weights:  21
## initial  value 23.760684
## iter  10 value 6.613531
## iter  20 value 6.312343
## final  value 6.312342
## converged
## # weights:  61
## initial  value 24.101719
## iter  10 value 4.070606
## iter  20 value 3.745704
## final  value 3.745688
## converged
## # weights:  101
## initial  value 22.512553
## iter  10 value 3.241565
## iter  20 value 3.170934
## iter  30 value 3.170690
## iter  30 value 3.170690
## iter  30 value 3.170690
## final  value 3.170690
## converged
## # weights:  21
## initial  value 22.428359
## iter  10 value 3.911001
## iter  20 value 3.900722
## iter  30 value 0.093798
## iter  40 value 0.056549
## iter  50 value 0.051296
## iter  60 value 0.044678
## iter  70 value 0.043637
## iter  80 value 0.043624
## iter  90 value 0.043616
## iter 100 value 0.043612
## final  value 0.043612
## stopped after 100 iterations
## # weights:  61
## initial  value 25.340091
## iter  10 value 0.090587
## iter  20 value 0.024863
```

```
## iter  30 value 0.022739
## iter  40 value 0.020063
## iter  50 value 0.018882
## iter  60 value 0.018353
## iter  70 value 0.017742
## iter  80 value 0.017571
## iter  90 value 0.017476
## iter 100 value 0.017429
## final  value 0.017429
## stopped after 100 iterations
## # weights:  101
## initial  value 25.850970
## iter  10 value 0.153701
## iter  20 value 0.037424
## iter  30 value 0.031458
## iter  40 value 0.023860
## iter  50 value 0.017330
## iter  60 value 0.016098
## iter  70 value 0.015158
## iter  80 value 0.014860
## iter  90 value 0.013607
## iter 100 value 0.013260
## final  value 0.013260
## stopped after 100 iterations
## # weights:  21
## initial  value 30.968578
## iter  10 value 0.016240
## final  value 0.000080
## converged
## # weights:  61
## initial  value 20.968870
## final  value 0.000073
## converged
## # weights:  101
## initial  value 21.650673
## final  value 0.000072
## converged
## # weights:  21
## initial  value 23.034467
## iter  10 value 6.081388
## iter  20 value 6.078852
## iter  20 value 6.078852
## iter  20 value 6.078852
## final  value 6.078852
## converged
## # weights:  61
## initial  value 33.572898
## iter  10 value 3.662935
## iter  20 value 3.654340
## final  value 3.654340
```

```
## converged
## # weights:  101
## initial  value 24.781850
## iter  10 value 3.509989
## iter  20 value 3.066044
## final  value 3.065757
## converged
## # weights:  21
## initial  value 23.439196
## iter  10 value 0.106797
## iter  20 value 0.059310
## iter  30 value 0.055548
## iter  40 value 0.045870
## iter  50 value 0.044928
## iter  60 value 0.044175
## iter  70 value 0.043878
## iter  80 value 0.043680
## iter  90 value 0.043666
## iter 100 value 0.043663
## final  value 0.043663
## stopped after 100 iterations
## # weights:  61
## initial  value 22.200206
## iter  10 value 0.156045
## iter  20 value 0.114078
## iter  30 value 0.053080
## iter  40 value 0.039932
## iter  50 value 0.030815
## iter  60 value 0.026548
## iter  70 value 0.025262
## iter  80 value 0.021260
## iter  90 value 0.019525
## iter 100 value 0.018746
## final  value 0.018746
## stopped after 100 iterations
## # weights:  101
## initial  value 21.600997
## iter  10 value 0.044281
## iter  20 value 0.030583
## iter  30 value 0.020383
## iter  40 value 0.017061
## iter  50 value 0.016312
## iter  60 value 0.014683
## iter  70 value 0.014327
## iter  80 value 0.014152
## iter  90 value 0.013845
## iter 100 value 0.013807
## final  value 0.013807
## stopped after 100 iterations
## # weights:  21
```

```
## initial  value 24.854893
## iter  10 value 8.354316
## iter  20 value 7.922365
## final  value 7.921987
## converged
## # weights:  61
## initial  value 21.278116
## iter  10 value 0.028781
## final  value 0.000081
## converged
## # weights:  101
## initial  value 27.593175
## iter  10 value 0.000492
## final  value 0.000070
## converged
## # weights:  21
## initial  value 23.212509
## iter  10 value 6.380459
## iter  20 value 6.309674
## final  value 6.309674
## converged
## # weights:  61
## initial  value 24.363645
## iter  10 value 5.672117
## iter  20 value 4.826287
## iter  30 value 4.805305
## iter  40 value 4.805028
## final  value 4.805027
## converged
## # weights:  101
## initial  value 25.431127
## iter  10 value 3.491059
## iter  20 value 3.330465
## final  value 3.330459
## converged
## # weights:  21
## initial  value 22.324237
## iter  10 value 7.976768
## iter  20 value 7.955077
## iter  30 value 7.947351
## iter  40 value 7.943310
## iter  50 value 7.942624
## iter  60 value 0.171539
## iter  70 value 0.070865
## iter  80 value 0.062618
## iter  90 value 0.046773
## iter 100 value 0.044442
## final  value 0.044442
## stopped after 100 iterations
## # weights:  61
```

```
## initial  value 22.633325
## iter  10 value 0.053290
## iter  20 value 0.044536
## iter  30 value 0.036175
## iter  40 value 0.033382
## iter  50 value 0.030079
## iter  60 value 0.028539
## iter  70 value 0.026946
## iter  80 value 0.026258
## iter  90 value 0.026085
## iter 100 value 0.025787
## final  value 0.025787
## stopped after 100 iterations
## # weights:  101
## initial  value 20.183515
## iter  10 value 0.037961
## iter  20 value 0.033059
## iter  30 value 0.027453
## iter  40 value 0.021212
## iter  50 value 0.018723
## iter  60 value 0.018250
## iter  70 value 0.017244
## iter  80 value 0.016533
## iter  90 value 0.016010
## iter 100 value 0.015848
## final  value 0.015848
## stopped after 100 iterations
## # weights:  21
## initial  value 24.044107
## iter  10 value 6.658771
## iter  20 value 0.009710
## final  value 0.000054
## converged
## # weights:  61
## initial  value 20.079808
## iter  10 value 0.006420
## final  value 0.000061
## converged
## # weights:  101
## initial  value 24.752066
## iter  10 value 0.001186
## iter  20 value 0.000695
## final  value 0.000093
## converged
## # weights:  21
## initial  value 23.416495
## iter  10 value 6.360017
## iter  20 value 6.331070
## iter  20 value 6.331070
## iter  20 value 6.331070
```

```
## final  value 6.331070
## converged
## # weights:  61
## initial  value 30.482285
## iter  10 value 4.463281
## iter  20 value 3.856039
## iter  30 value 3.854156
## final  value 3.854156
## converged
## # weights:  101
## initial  value 28.555645
## iter  10 value 3.713784
## iter  20 value 3.567541
## iter  30 value 3.567384
## final  value 3.567383
## converged
## # weights:  21
## initial  value 24.668911
## iter  10 value 0.075040
## iter  20 value 0.057941
## iter  30 value 0.052584
## iter  40 value 0.044381
## iter  50 value 0.044050
## iter  60 value 0.044033
## iter  70 value 0.043994
## iter  80 value 0.043975
## iter  90 value 0.043972
## final  value 0.043971
## converged
## # weights:  61
## initial  value 23.836073
## iter  10 value 0.025515
## iter  20 value 0.022132
## iter  30 value 0.019389
## iter  40 value 0.019116
## iter  50 value 0.018448
## iter  60 value 0.017988
## iter  70 value 0.017785
## iter  80 value 0.017703
## iter  90 value 0.017684
## iter 100 value 0.017666
## final  value 0.017666
## stopped after 100 iterations
## # weights:  101
## initial  value 18.764264
## iter  10 value 0.027242
## iter  20 value 0.021602
## iter  30 value 0.018605
## iter  40 value 0.017209
## iter  50 value 0.016630
```

```
## iter  60 value 0.016205
## iter  70 value 0.015625
## iter  80 value 0.015146
## iter  90 value 0.014856
## iter 100 value 0.014773
## final  value 0.014773
## stopped after 100 iterations
## # weights:  21
## initial  value 25.613709
## iter  10 value 5.624962
## iter  20 value 3.636915
## iter  30 value 3.602512
## final  value 3.602461
## converged
## # weights:  61
## initial  value 20.579061
## iter  10 value 0.186144
## iter  20 value 0.000576
## final  value 0.000083
## converged
## # weights:  101
## initial  value 24.186752
## iter  10 value 0.022670
## final  value 0.000093
## converged
## # weights:  21
## initial  value 24.311449
## iter  10 value 6.418088
## iter  20 value 6.397502
## final  value 6.397502
## converged
## # weights:  61
## initial  value 23.984019
## iter  10 value 3.837133
## iter  20 value 3.778575
## final  value 3.778574
## converged
## # weights:  101
## initial  value 29.218506
## iter  10 value 3.326300
## iter  20 value 3.235457
## final  value 3.235354
## converged
## # weights:  21
## initial  value 20.919743
## iter  10 value 0.057515
## iter  20 value 0.046735
## iter  30 value 0.044394
## iter  40 value 0.044065
## iter  50 value 0.044042
```

```
## iter  60 value 0.044028
## iter  70 value 0.044023
## iter  80 value 0.044021
## iter  90 value 0.044020
## final  value 0.044020
## converged
## # weights:  61
## initial  value 20.350934
## iter  10 value 0.050231
## iter  20 value 0.035135
## iter  30 value 0.027645
## iter  40 value 0.023452
## iter  50 value 0.020679
## iter  60 value 0.020039
## iter  70 value 0.018976
## iter  80 value 0.018662
## iter  90 value 0.018034
## iter 100 value 0.017891
## final  value 0.017891
## stopped after 100 iterations
## # weights:  101
## initial  value 25.320215
## iter  10 value 0.062853
## iter  20 value 0.027051
## iter  30 value 0.024091
## iter  40 value 0.020366
## iter  50 value 0.018635
## iter  60 value 0.014917
## iter  70 value 0.013786
## iter  80 value 0.013551
## iter  90 value 0.012890
## iter 100 value 0.012802
## final  value 0.012802
## stopped after 100 iterations
## # weights:  21
## initial  value 23.778040
## iter  10 value 0.636737
## iter  20 value 0.001901
## final  value 0.000086
## converged
## # weights:  61
## initial  value 19.899114
## iter  10 value 0.006007
## final  value 0.000099
## converged
## # weights:  101
## initial  value 18.439960
## iter  10 value 0.000283
## final  value 0.000071
## converged
```

```
## # weights:  21
## initial  value 29.562832
## iter  10 value 7.839949
## iter  20 value 6.312539
## final  value 6.312321
## converged
## # weights:  61
## initial  value 23.119888
## iter  10 value 3.967161
## iter  20 value 3.748989
## final  value 3.748987
## converged
## # weights:  101
## initial  value 19.765822
## iter  10 value 3.611965
## iter  20 value 3.174222
## final  value 3.174045
## converged
## # weights:  21
## initial  value 24.136082
## iter  10 value 0.215101
## iter  20 value 0.052713
## iter  30 value 0.050215
## iter  40 value 0.048127
## iter  50 value 0.043404
## iter  60 value 0.043135
## iter  70 value 0.042724
## iter  80 value 0.042610
## iter  90 value 0.042518
## iter 100 value 0.042494
## final  value 0.042494
## stopped after 100 iterations
## # weights:  61
## initial  value 22.795826
## iter  10 value 4.302215
## iter  20 value 0.050281
## iter  30 value 0.045810
## iter  40 value 0.036253
## iter  50 value 0.033844
## iter  60 value 0.027353
## iter  70 value 0.022699
## iter  80 value 0.021606
## iter  90 value 0.018695
## iter 100 value 0.018357
## final  value 0.018357
## stopped after 100 iterations
## # weights:  101
## initial  value 21.848322
## iter  10 value 2.200752
## iter  20 value 0.128526
```

```
## iter  30 value 0.101861
## iter  40 value 0.083483
## iter  50 value 0.049608
## iter  60 value 0.038323
## iter  70 value 0.030222
## iter  80 value 0.022868
## iter  90 value 0.020234
## iter 100 value 0.019196
## final  value 0.019196
## stopped after 100 iterations
## # weights:  21
## initial  value 24.109944
## iter  10 value 3.675048
## iter  20 value 3.673977
## final  value 3.673950
## converged
## # weights:  61
## initial  value 22.473074
## iter  10 value 0.060109
## final  value 0.000068
## converged
## # weights:  101
## initial  value 21.959150
## iter  10 value 0.068633
## final  value 0.000068
## converged
## # weights:  21
## initial  value 24.441128
## iter  10 value 6.375552
## iter  20 value 6.315869
## iter  20 value 6.315869
## iter  20 value 6.315869
## final  value 6.315869
## converged
## # weights:  61
## initial  value 25.301348
## iter  10 value 3.987956
## iter  20 value 3.899966
## final  value 3.899965
## converged
## # weights:  101
## initial  value 23.266929
## iter  10 value 3.572024
## iter  20 value 3.342813
## final  value 3.341999
## converged
## # weights:  21
## initial  value 21.341375
## iter  10 value 3.705857
## iter  20 value 3.700581
```

```
## iter  30 value 0.188243
## iter  40 value 0.045864
## iter  50 value 0.045347
## iter  60 value 0.044347
## iter  70 value 0.044125
## iter  80 value 0.044092
## iter  90 value 0.044080
## iter 100 value 0.044074
## final  value 0.044074
## stopped after 100 iterations
## # weights:  61
## initial  value 28.285462
## iter  10 value 0.091585
## iter  20 value 0.056318
## iter  30 value 0.036774
## iter  40 value 0.029647
## iter  50 value 0.027438
## iter  60 value 0.023579
## iter  70 value 0.020601
## iter  80 value 0.018732
## iter  90 value 0.018421
## iter 100 value 0.018296
## final  value 0.018296
## stopped after 100 iterations
## # weights:  101
## initial  value 24.041973
## iter  10 value 0.042023
## iter  20 value 0.029203
## iter  30 value 0.020869
## iter  40 value 0.018091
## iter  50 value 0.016844
## iter  60 value 0.015277
## iter  70 value 0.014508
## iter  80 value 0.013757
## iter  90 value 0.013361
## iter 100 value 0.013217
## final  value 0.013217
## stopped after 100 iterations
## # weights:  21
## initial  value 21.421158
## iter  10 value 0.006407
## final  value 0.000051
## converged
## # weights:  61
## initial  value 22.093244
## final  value 0.000057
## converged
## # weights:  101
## initial  value 30.375469
## final  value 0.000064
```

```
## converged
## # weights:  21
## initial  value 22.094364
## iter  10 value 6.171668
## final  value 6.170070
## converged
## # weights:  61
## initial  value 23.108705
## iter  10 value 4.059486
## iter  20 value 3.618887
## final  value 3.618859
## converged
## # weights:  101
## initial  value 28.468634
## iter  10 value 3.085527
## iter  20 value 3.038428
## final  value 3.038378
## converged
## # weights:  21
## initial  value 22.175620
## iter  10 value 0.074462
## iter  20 value 0.045203
## iter  30 value 0.044805
## iter  40 value 0.044675
## iter  50 value 0.043957
## iter  60 value 0.043809
## iter  70 value 0.043723
## iter  80 value 0.043670
## iter  90 value 0.043665
## iter 100 value 0.043665
## final  value 0.043665
## stopped after 100 iterations
## # weights:  61
## initial  value 23.195835
## iter  10 value 0.028236
## iter  20 value 0.024514
## iter  30 value 0.020671
## iter  40 value 0.018821
## iter  50 value 0.017990
## iter  60 value 0.017740
## iter  70 value 0.017581
## iter  80 value 0.017448
## iter  90 value 0.017295
## iter 100 value 0.017204
## final  value 0.017204
## stopped after 100 iterations
## # weights:  101
## initial  value 19.762800
## iter  10 value 0.039050
## iter  20 value 0.029409
```

```
## iter  30 value 0.026029
## iter  40 value 0.019255
## iter  50 value 0.016346
## iter  60 value 0.014768
## iter  70 value 0.014442
## iter  80 value 0.014109
## iter  90 value 0.013983
## iter 100 value 0.013965
## final  value 0.013965
## stopped after 100 iterations
## # weights:  21
## initial  value 22.171335
## iter  10 value 0.010844
## final  value 0.000088
## converged
## # weights:  61
## initial  value 22.164441
## iter  10 value 0.012844
## final  value 0.000053
## converged
## # weights:  101
## initial  value 25.400699
## iter  10 value 0.000412
## final  value 0.000063
## converged
## # weights:  21
## initial  value 22.899509
## iter  10 value 6.253026
## iter  20 value 6.236335
## final  value 6.236335
## converged
## # weights:  61
## initial  value 25.833198
## iter  10 value 4.672648
## iter  20 value 4.668175
## final  value 4.668173
## converged
## # weights:  101
## initial  value 25.185509
## iter  10 value 3.238324
## iter  20 value 3.128292
## final  value 3.128260
## converged
## # weights:  21
## initial  value 21.939404
## iter  10 value 0.329297
## iter  20 value 0.098686
## iter  30 value 0.071659
## iter  40 value 0.059524
## iter  50 value 0.051911
```

```
## iter  60 value 0.048228
## iter  70 value 0.043891
## iter  80 value 0.043740
## iter  90 value 0.043728
## iter 100 value 0.043725
## final  value 0.043725
## stopped after 100 iterations
## # weights:  61
## initial  value 22.993222
## iter  10 value 0.047837
## iter  20 value 0.035295
## iter  30 value 0.028391
## iter  40 value 0.023734
## iter  50 value 0.019677
## iter  60 value 0.018623
## iter  70 value 0.017764
## iter  80 value 0.017490
## iter  90 value 0.017335
## iter 100 value 0.017298
## final  value 0.017298
## stopped after 100 iterations
## # weights:  101
## initial  value 21.504930
## iter  10 value 0.019444
## iter  20 value 0.016202
## iter  30 value 0.014014
## iter  40 value 0.013076
## iter  50 value 0.012661
## iter  60 value 0.012528
## iter  70 value 0.012476
## iter  80 value 0.012442
## iter  90 value 0.012422
## iter 100 value 0.012420
## final  value 0.012420
## stopped after 100 iterations
## # weights:  21
## initial  value 27.036142
## final  value 0.000000
## converged
## # weights:  61
## initial  value 20.961481
## final  value 0.000006
## converged
## # weights:  101
## initial  value 28.402957
## iter  10 value 0.003440
## final  value 0.000054
## converged
## # weights:  21
## initial  value 27.117709
```

```
## iter  10 value 6.469509
## iter  20 value 6.183331
## final  value 6.183331
## converged
## # weights:  61
## initial  value 23.652120
## iter  10 value 3.698702
## iter  20 value 3.587346
## final  value 3.587345
## converged
## # weights:  101
## initial  value 27.882333
## iter  10 value 3.060043
## iter  20 value 2.994125
## final  value 2.994081
## converged
## # weights:  21
## initial  value 22.426012
## iter  10 value 0.066389
## iter  20 value 0.052084
## iter  30 value 0.047379
## iter  40 value 0.044127
## iter  50 value 0.043952
## iter  60 value 0.043647
## iter  70 value 0.043519
## iter  80 value 0.043495
## iter  90 value 0.043493
## final  value 0.043492
## converged
## # weights:  61
## initial  value 23.373742
## iter  10 value 0.049789
## iter  20 value 0.034441
## iter  30 value 0.031025
## iter  40 value 0.026116
## iter  50 value 0.024346
## iter  60 value 0.021374
## iter  70 value 0.018653
## iter  80 value 0.018210
## iter  90 value 0.017398
## iter 100 value 0.017223
## final  value 0.017223
## stopped after 100 iterations
## # weights:  101
## initial  value 24.969020
## iter  10 value 0.017665
## iter  20 value 0.015308
## iter  30 value 0.013622
## iter  40 value 0.012994
## iter  50 value 0.012273
```

```
## iter  60 value 0.012161
## iter  70 value 0.012106
## iter  80 value 0.012102
## iter  90 value 0.012101
## iter 100 value 0.012101
## final  value 0.012101
## stopped after 100 iterations
## # weights:  21
## initial  value 23.782805
## iter  10 value 6.397178
## final  value 6.387129
## converged
```

```
varImp(modelFitd)
```

```
## nnet variable importance
##
##                Overall
## inner_TGFB1   100.000
## inner_ALPL     71.732
## inner_OCT4     62.601
## inner_PDGFB    52.400
## inner_VWF      50.339
## inner_Col10A1  49.077
## inner_BMP2     40.203
## inner_Osterix  30.755
## inner_PPARG    20.369
## inner_Col2A1   19.593
## inner_EGFR     19.129
## inner_BMP6     18.139
## inner_PTHR     16.448
## inner_IBSP     15.875
## inner_NANOG    12.339
## inner_Sox2      9.998
## inner_Col1A1    8.083
## inner_GLA       0.000
```

```
# DO the same for the outer section

##############################################################################################
#                                                OUTER SECTION
##############################################################################################
#  IM outer, perform data transformation

preObj1 <- preProcess(goat_IM[,c(8,41:59)], method=c("scale","center"))

# transform the dataset using the parameters
transformed1 <- predict(preObj1, goat_IM[,c(8,41:59)])

##############################################################################################
# Look for genes that are importante in spacer texture
```

```
###################################
# another way of looking for important genes using the popular data analytics techniques
#########################################################
# Boruta
Boruta(Spacer.texture~.,data=transformed1,doTrace=2)->Bor.imbio1

##  1. run of importance source...

##  2. run of importance source...

##  3. run of importance source...

##  4. run of importance source...

##  5. run of importance source...

##  6. run of importance source...

##  7. run of importance source...

##  8. run of importance source...

##  9. run of importance source...

##  10. run of importance source...

##  11. run of importance source...

## Confirmed 11 attributes: outer_ALPL, outer_BMP2, outer_BMP6, outer_Col1A1, out
er_GLA and 6 more.

## Rejected 3 attributes: outer_Col10A1, outer_Col2A1, outer_PPARG.

##  12. run of importance source...

##  13. run of importance source...

##  14. run of importance source...

##  15. run of importance source...

## Confirmed 2 attributes: outer_NANOG, outer_PTHR.

##  16. run of importance source...

##  17. run of importance source...

##  18. run of importance source...

##  19. run of importance source...

## Confirmed 1 attributes: outer_OCT4.

## Rejected 1 attributes: outer_Sox2.

##  20. run of importance source...
```

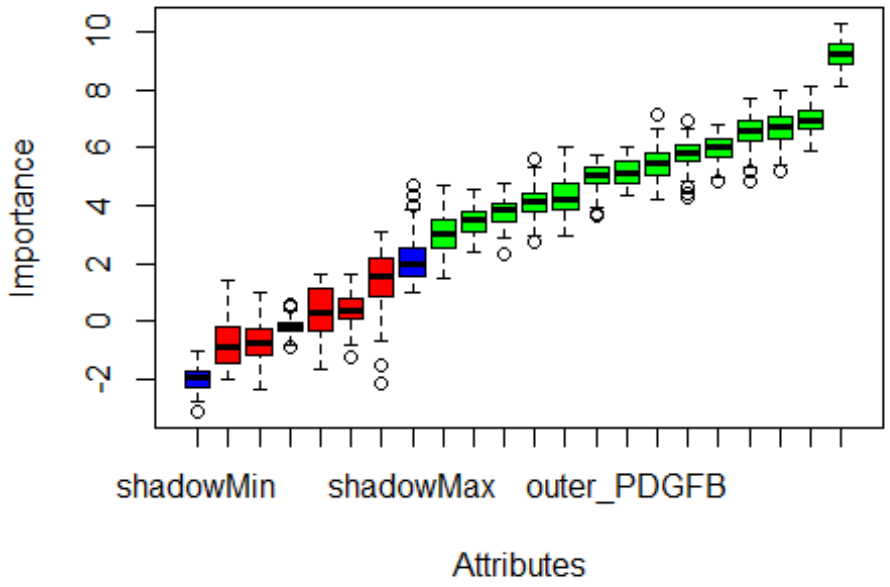
```
##  21. run of importance source...
##  22. run of importance source...
##  23. run of importance source...
##  24. run of importance source...
##  25. run of importance source...
##  26. run of importance source...
##  27. run of importance source...
##  28. run of importance source...
##  29. run of importance source...
##  30. run of importance source...
##  31. run of importance source...
##  32. run of importance source...
##  33. run of importance source...
##  34. run of importance source...
##  35. run of importance source...
##  36. run of importance source...
##  37. run of importance source...
##  38. run of importance source...
##  39. run of importance source...
##  40. run of importance source...
##  41. run of importance source...
##  42. run of importance source...
##  43. run of importance source...
##  44. run of importance source...
##  45. run of importance source...
##  46. run of importance source...
##  47. run of importance source...
##  48. run of importance source...
##  49. run of importance source...
##  50. run of importance source...
```

```
##  51. run of importance source...
##  52. run of importance source...
##  53. run of importance source...
##  54. run of importance source...
##  55. run of importance source...
##  56. run of importance source...
##  57. run of importance source...
##  58. run of importance source...
##  59. run of importance source...
##  60. run of importance source...
##  61. run of importance source...
##  62. run of importance source...
##  63. run of importance source...
##  64. run of importance source...
##  65. run of importance source...
##  66. run of importance source...
##  67. run of importance source...
##  68. run of importance source...
##  69. run of importance source...
##  70. run of importance source...
##  71. run of importance source...
##  72. run of importance source...
##  73. run of importance source...
## Rejected 1 attributes: outer_EGFR.
```

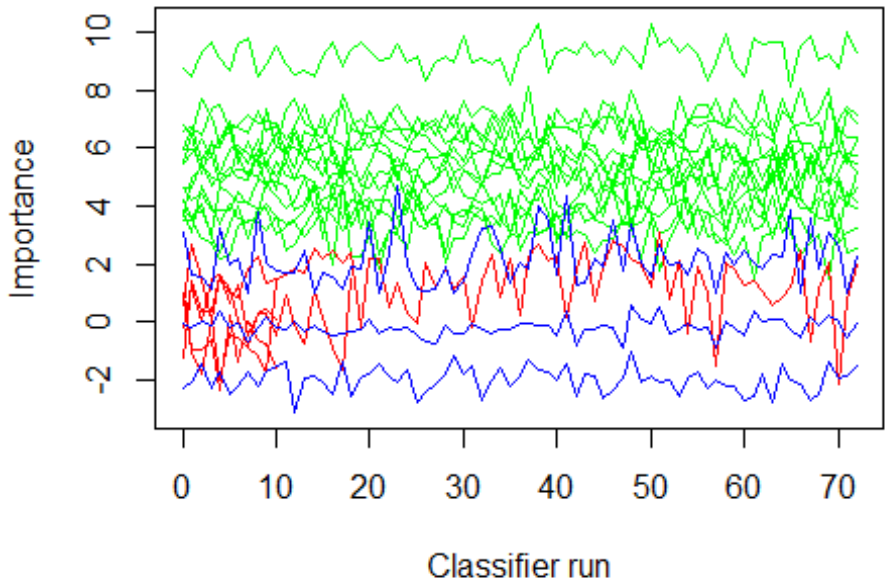
```
print(Bor.imbio1,zero.print=".")
```

```
## Boruta performed 73 iterations in 1.849179 secs.
##  14 attributes confirmed important: outer_ALPL, outer_BMP2,
## outer_BMP6, outer_Col1A1, outer_GLA and 9 more.
##  5 attributes confirmed unimportant: outer_Col10A1, outer_Col2A1,
## outer_EGFR, outer_PPARG, outer_Sox2.
```

```
plot(Bor.imbio1)
```




```
plotImpHistory(Bor.imbio1)
```

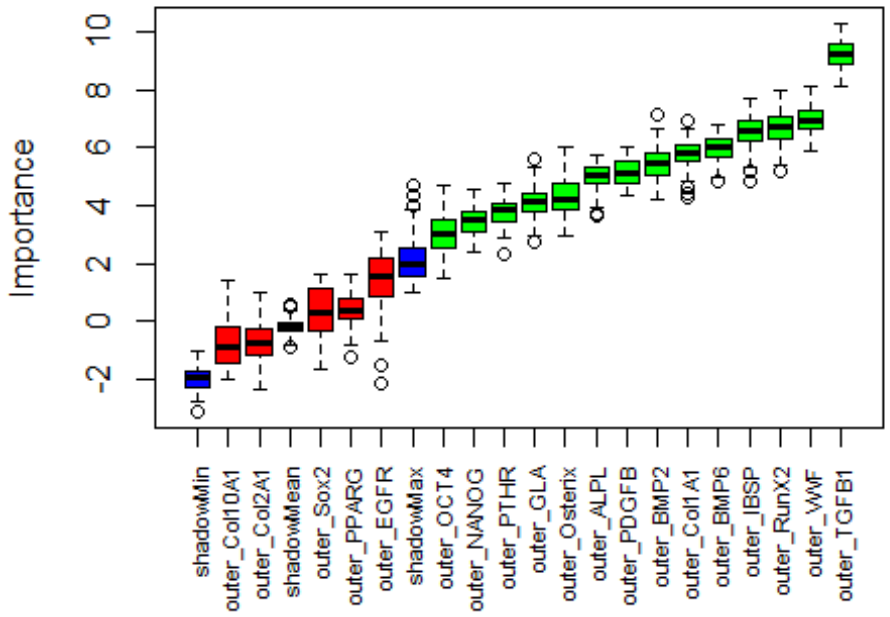



```
attStats(Bor.imbio1)
```

```
##                  meanImp  medianImp    minImp    maxImp   normHits
## outer_RunX2    6.6886880  6.7242385  5.166618  7.991757 1.00000000
## outer_Osterix  4.3036266  4.2189961  2.949320  5.998989 0.95890411
## outer_PPARG    0.3558997  0.3557501 -1.211721  1.612903 0.00000000
```

```
## outer_OCT4     3.0069791  3.0006826  1.459754  4.724005 0.78082192
## outer_Sox2     0.3141160  0.2795173 -1.654207  1.649745 0.02739726
## outer_NANOG    3.4757992  3.5008650  2.386417  4.582524 0.87671233
## outer_ALPL     4.9937629  5.0690656  3.628647  5.733896 1.00000000
## outer_EGFR     1.4217344  1.5969400 -2.159112  3.076118 0.30136986
## outer_VWF      6.9575207  6.9258721  5.875399  8.113045 1.00000000
## outer_PDGFB    5.1674654  5.1524870  4.370011  5.994051 1.00000000
## outer_TGFB1    9.2002164  9.2269751  8.093976 10.290159 1.00000000
## outer_BMP2     5.4276811  5.4423695  4.202102  7.122480 1.00000000
## outer_BMP6     5.9971060  6.0250442  4.814705  6.769152 1.00000000
## outer_Col1A1   5.7817140  5.8255155  4.302380  6.939787 1.00000000
## outer_Col2A1  -0.7139071 -0.7117970 -2.372963  1.018020 0.00000000
## outer_Col10A1 -0.6335768 -0.9016173 -1.975548  1.398899 0.00000000
## outer_GLA      4.1547532  4.1537226  2.732838  5.607439 0.95890411
## outer_IBSP     6.5635198  6.6116803  4.822855  7.734858 1.00000000
## outer_PTHR     3.8148200  3.8555714  2.337458  4.806678 0.94520548
##                 decision
## outer_RunX2    Confirmed
## outer_Osterix  Confirmed
## outer_PPARG     Rejected
## outer_OCT4     Confirmed
## outer_Sox2      Rejected
## outer_NANOG    Confirmed
## outer_ALPL     Confirmed
## outer_EGFR      Rejected
## outer_VWF      Confirmed
## outer_PDGFB    Confirmed
## outer_TGFB1    Confirmed
## outer_BMP2     Confirmed
## outer_BMP6     Confirmed
## outer_Col1A1   Confirmed
## outer_Col2A1    Rejected
## outer_Col10A1   Rejected
## outer_GLA      Confirmed
## outer_IBSP     Confirmed
## outer_PTHR     Confirmed
```

```r
plot(Bor.imbio1, xlab = "", xaxt = "n")
lz<-lapply(1:ncol(Bor.imbio1$ImpHistory),function(i)
  Bor.imbio1$ImpHistory[is.finite(Bor.imbio1$ImpHistory[,i]),i])
names(lz) <- colnames(Bor.imbio1$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(side = 1,las=2,labels = names(Labels),
     at = 1:ncol(Bor.imbio1$ImpHistory), cex.axis = 0.7)
```




```
#Random Forest
modelFit1 <- train(Spacer.texture~.,data=transformed1, method="rf" )  # random forest:  g
enes are ranked from most important to least
varImp(modelFit1)

## rf variable importance
##
##                Overall
## outer_TGFB1   100.000
## outer_IBSP     68.784
## outer_RunX2    56.851
## outer_ALPL     54.227
## outer_BMP6     54.183
## outer_VWF      53.593
## outer_Col1A1   53.162
## outer_Osterix  45.702
## outer_PDGFB    38.631
## outer_BMP2     36.108
## outer_OCT4     31.265
## outer_GLA      28.566
## outer_NANOG    26.885
## outer_PTHR     24.170
## outer_EGFR     19.119
## outer_Col2A1   12.470
## outer_Sox2      6.861
## outer_PPARG     2.459
## outer_Col10A1   0.000
```

```
######################
modelFit2 <- train(Spacer.texture~.,data=transformed1, method="glmnet" )
varImp(modelFit2)

## glmnet variable importance
##
##               Overall
## outer_TGFB1   100.000
## outer_BMP6     17.679
## outer_ALPL     11.890
## outer_Sox2      7.547
## outer_PPARG     6.570
## outer_EGFR      0.000
## outer_Col2A1    0.000
## outer_RunX2     0.000
## outer_NANOG     0.000
## outer_Osterix   0.000
## outer_PTHR      0.000
## outer_Col1A1    0.000
## outer_BMP2      0.000
## outer_GLA       0.000
## outer_IBSP      0.000
## outer_VWF       0.000
## outer_OCT4      0.000
## outer_Col10A1   0.000
## outer_PDGFB     0.000

#####################################################################
modelFit3 <- train(Spacer.texture~.,data=transformed1, method="pda" )

## Warning: predictions failed for Resample01: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning: predictions failed for Resample06: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning: predictions failed for Resample07: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning: predictions failed for Resample10: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning: predictions failed for Resample12: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning: predictions failed for Resample13: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments
```

```
## Warning: predictions failed for Resample14: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning: predictions failed for Resample16: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning: predictions failed for Resample19: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning: predictions failed for Resample21: lambda=0e+00 Error in mindist[l] <
- ndist[l] :
##   NAs are not allowed in subscripted assignments

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.

varImp(modelFit3)

## ROC curve variable importance
##
##               Importance
## outer_TGFB1       100.00
## outer_BMP6         98.39
## outer_VWF          96.77
## outer_BMP2         95.16
## outer_Col1A1       94.62
## outer_RunX2        93.01
## outer_IBSP         92.47
## outer_PTHR         92.47
## outer_Osterix      91.94
## outer_GLA          91.94
## outer_ALPL         91.94
## outer_PDGFB        90.32
## outer_EGFR         86.56
## outer_OCT4         83.33
## outer_NANOG        82.26
## outer_Col2A1       65.59
## outer_PPARG        54.30
## outer_Col10A1      42.47
## outer_Sox2          0.00

#################################################################
modelFit4 <- train(Spacer.texture~.,data=transformed1, method="nnet" )

## # weights:  22
## initial  value 19.350389
## iter  10 value 3.250830
## iter  10 value 3.250830
## iter  10 value 3.250830
## final  value 3.250830
```

```
## converged
## # weights:  64
## initial  value 36.968166
## iter  10 value 3.261772
## iter  20 value 3.251054
## iter  30 value 3.250830
## final  value 3.250830
## converged
## # weights:  106
## initial  value 23.159652
## iter  10 value 0.025824
## final  value 0.000069
## converged
## # weights:  22
## initial  value 19.141820
## iter  10 value 6.479664
## iter  20 value 6.150274
## final  value 6.150269
## converged
## # weights:  64
## initial  value 26.589721
## iter  10 value 4.750492
## iter  20 value 4.076745
## final  value 4.076589
## converged
## # weights:  106
## initial  value 26.410259
## iter  10 value 4.083351
## iter  20 value 3.813005
## iter  30 value 3.800039
## final  value 3.799938
## converged
## # weights:  22
## initial  value 20.969678
## iter  10 value 3.288157
## iter  20 value 3.281050
## iter  30 value 0.104401
## iter  40 value 0.050595
## iter  50 value 0.049364
## iter  60 value 0.045547
## iter  70 value 0.045110
## iter  80 value 0.044735
## iter  90 value 0.044648
## iter 100 value 0.044612
## final  value 0.044612
## stopped after 100 iterations
## # weights:  64
## initial  value 25.995222
## iter  10 value 3.291189
## iter  20 value 3.280057
```

```
## iter  30 value 1.785381
## iter  40 value 0.056246
## iter  50 value 0.042289
## iter  60 value 0.033424
## iter  70 value 0.027898
## iter  80 value 0.027284
## iter  90 value 0.022441
## iter 100 value 0.021802
## final  value 0.021802
## stopped after 100 iterations
## # weights:  106
## initial  value 19.873139
## iter  10 value 0.063668
## iter  20 value 0.038793
## iter  30 value 0.033238
## iter  40 value 0.029284
## iter  50 value 0.027800
## iter  60 value 0.025020
## iter  70 value 0.022253
## iter  80 value 0.019042
## iter  90 value 0.017971
## iter 100 value 0.017268
## final  value 0.017268
## stopped after 100 iterations
## # weights:  22
## initial  value 22.240969
## iter  10 value 4.272534
## iter  20 value 3.918286
## iter  30 value 3.917652
## final  value 3.917651
## converged
## # weights:  64
## initial  value 23.043844
## iter  10 value 0.690673
## iter  20 value 0.002019
## final  value 0.000089
## converged
## # weights:  106
## initial  value 26.638569
## iter  10 value 0.022008
## iter  20 value 0.000186
## iter  20 value 0.000094
## iter  20 value 0.000094
## final  value 0.000094
## converged
## # weights:  22
## initial  value 22.496536
## iter  10 value 9.932732
## iter  20 value 9.733884
## final  value 9.733883
```

```
## converged
## # weights:  64
## initial  value 25.461316
## iter  10 value 5.783706
## iter  20 value 5.610964
## final  value 5.610768
## converged
## # weights:  106
## initial  value 21.829632
## iter  10 value 5.652269
## iter  20 value 5.123857
## iter  30 value 5.109114
## final  value 5.109108
## converged
## # weights:  22
## initial  value 22.827784
## iter  10 value 7.488298
## iter  20 value 7.480290
## iter  30 value 7.473369
## iter  40 value 3.963350
## iter  50 value 3.955553
## iter  60 value 3.953745
## iter  70 value 3.953233
## iter  80 value 3.953154
## iter  90 value 3.953127
## iter 100 value 3.953107
## final  value 3.953107
## stopped after 100 iterations
## # weights:  64
## initial  value 23.774684
## iter  10 value 4.810651
## iter  20 value 2.469356
## iter  30 value 0.342639
## iter  40 value 0.262939
## iter  50 value 0.219659
## iter  60 value 0.175027
## iter  70 value 0.141112
## iter  80 value 0.117491
## iter  90 value 0.087304
## iter 100 value 0.064764
## final  value 0.064764
## stopped after 100 iterations
## # weights:  106
## initial  value 26.875011
## iter  10 value 0.143334
## iter  20 value 0.079633
## iter  30 value 0.059516
## iter  40 value 0.050171
## iter  50 value 0.042960
## iter  60 value 0.041581
```

```
## iter  70 value 0.037984
## iter  80 value 0.036612
## iter  90 value 0.034979
## iter 100 value 0.032623
## final  value 0.032623
## stopped after 100 iterations
## # weights:  22
## initial  value 23.254136
## iter  10 value 3.684148
## iter  20 value 3.480027
## iter  30 value 0.047915
## iter  40 value 0.006970
## final  value 0.000098
## converged
## # weights:  64
## initial  value 20.917685
## iter  10 value 1.111591
## iter  20 value 0.000870
## final  value 0.000083
## converged
## # weights:  106
## initial  value 24.836861
## iter  10 value 0.280382
## iter  20 value 0.000285
## final  value 0.000074
## converged
## # weights:  22
## initial  value 22.421534
## iter  10 value 7.121067
## iter  20 value 6.985173
## final  value 6.985171
## converged
## # weights:  64
## initial  value 23.117846
## iter  10 value 4.932319
## iter  20 value 4.754607
## final  value 4.754589
## converged
## # weights:  106
## initial  value 28.800755
## iter  10 value 4.375918
## iter  20 value 4.273990
## iter  30 value 4.273838
## final  value 4.273838
## converged
## # weights:  22
## initial  value 24.280670
## iter  10 value 3.675968
## iter  20 value 0.491762
## iter  30 value 0.049854
```

```
## iter  40 value 0.048424
## iter  50 value 0.047934
## iter  60 value 0.047842
## iter  70 value 0.047627
## iter  80 value 0.047578
## iter  90 value 0.047570
## iter 100 value 0.047566
## final  value 0.047566
## stopped after 100 iterations
## # weights:  64
## initial  value 22.907326
## iter  10 value 0.076555
## iter  20 value 0.036789
## iter  30 value 0.031620
## iter  40 value 0.028791
## iter  50 value 0.026123
## iter  60 value 0.025170
## iter  70 value 0.024647
## iter  80 value 0.023771
## iter  90 value 0.022968
## iter 100 value 0.022752
## final  value 0.022752
## stopped after 100 iterations
## # weights:  106
## initial  value 23.535573
## iter  10 value 3.283702
## iter  20 value 0.045994
## iter  30 value 0.036615
## iter  40 value 0.031650
## iter  50 value 0.023591
## iter  60 value 0.021977
## iter  70 value 0.020338
## iter  80 value 0.019914
## iter  90 value 0.019346
## iter 100 value 0.019038
## final  value 0.019038
## stopped after 100 iterations
## # weights:  22
## initial  value 20.476224
## final  value 0.000094
## converged
## # weights:  64
## initial  value 19.824514
## iter  10 value 7.277359
## iter  20 value 0.153548
## iter  30 value 0.000140
## iter  30 value 0.000072
## iter  30 value 0.000072
## final  value 0.000072
## converged
```

```
## # weights:  106
## initial  value 21.760161
## final  value 0.000000
## converged
## # weights:  22
## initial  value 24.783979
## iter  10 value 6.535501
## iter  20 value 6.263253
## final  value 6.263253
## converged
## # weights:  64
## initial  value 23.711148
## iter  10 value 6.141420
## iter  20 value 5.321617
## iter  30 value 5.317011
## final  value 5.317011
## converged
## # weights:  106
## initial  value 24.675607
## iter  10 value 3.925777
## iter  20 value 3.540524
## iter  30 value 3.533025
## final  value 3.533025
## converged
## # weights:  22
## initial  value 23.037767
## iter  10 value 7.304147
## iter  20 value 7.292084
## iter  30 value 7.291116
## iter  40 value 7.290800
## iter  50 value 7.290734
## iter  60 value 7.290692
## iter  70 value 7.290649
## iter  80 value 7.288289
## iter  90 value 0.140191
## iter 100 value 0.048525
## final  value 0.048525
## stopped after 100 iterations
## # weights:  64
## initial  value 21.884630
## iter  10 value 0.086842
## iter  20 value 0.057041
## iter  30 value 0.038087
## iter  40 value 0.036977
## iter  50 value 0.031257
## iter  60 value 0.029077
## iter  70 value 0.027824
## iter  80 value 0.027079
## iter  90 value 0.026469
## iter 100 value 0.025986
```

```
## final  value 0.025986
## stopped after 100 iterations
## # weights:  106
## initial  value 26.755716
## iter  10 value 0.124759
## iter  20 value 0.076476
## iter  30 value 0.045733
## iter  40 value 0.037311
## iter  50 value 0.036199
## iter  60 value 0.031571
## iter  70 value 0.025686
## iter  80 value 0.022271
## iter  90 value 0.019518
## iter 100 value 0.017383
## final  value 0.017383
## stopped after 100 iterations
## # weights:  22
## initial  value 21.749993
## iter  10 value 3.681913
## iter  20 value 3.673973
## final  value 3.673950
## converged
## # weights:  64
## initial  value 22.408019
## iter  10 value 0.291393
## iter  20 value 0.004166
## final  value 0.000066
## converged
## # weights:  106
## initial  value 24.523557
## iter  10 value 0.022133
## final  value 0.000061
## converged
## # weights:  22
## initial  value 22.504711
## iter  10 value 6.762248
## iter  20 value 6.377601
## final  value 6.377599
## converged
## # weights:  64
## initial  value 21.714544
## iter  10 value 4.155074
## iter  20 value 4.014252
## final  value 4.007674
## converged
## # weights:  106
## initial  value 28.870567
## iter  10 value 4.449708
## iter  20 value 3.458448
## iter  30 value 3.457206
```

```
## final  value 3.457206
## converged
## # weights:  22
## initial  value 21.906274
## iter  10 value 3.835223
## iter  20 value 3.799423
## iter  30 value 3.706647
## iter  40 value 3.702352
## iter  50 value 3.700475
## iter  60 value 3.698141
## iter  70 value 3.695099
## iter  80 value 3.671990
## iter  90 value 0.086809
## iter 100 value 0.051558
## final  value 0.051558
## stopped after 100 iterations
## # weights:  64
## initial  value 20.917169
## iter  10 value 3.765880
## iter  20 value 3.712066
## iter  30 value 3.696222
## iter  40 value 0.505686
## iter  50 value 0.050502
## iter  60 value 0.045488
## iter  70 value 0.039229
## iter  80 value 0.038124
## iter  90 value 0.037406
## iter 100 value 0.035307
## final  value 0.035307
## stopped after 100 iterations
## # weights:  106
## initial  value 21.276669
## iter  10 value 0.726112
## iter  20 value 0.032191
## iter  30 value 0.030001
## iter  40 value 0.022005
## iter  50 value 0.021110
## iter  60 value 0.018628
## iter  70 value 0.016546
## iter  80 value 0.015471
## iter  90 value 0.014725
## iter 100 value 0.013877
## final  value 0.013877
## stopped after 100 iterations
## # weights:  22
## initial  value 22.987583
## iter  10 value 0.147743
## iter  20 value 0.002634
## iter  30 value 0.000264
## final  value 0.000088
```

```
## converged
## # weights:  64
## initial  value 20.774501
## iter  10 value 0.028677
## final  value 0.000091
## converged
## # weights:  106
## initial  value 23.128257
## iter  10 value 3.015608
## iter  20 value 0.015231
## final  value 0.000064
## converged
## # weights:  22
## initial  value 22.619134
## iter  10 value 7.328602
## iter  20 value 7.289625
## iter  20 value 7.289625
## iter  20 value 7.289625
## final  value 7.289625
## converged
## # weights:  64
## initial  value 25.945532
## iter  10 value 5.071513
## iter  20 value 4.951767
## final  value 4.951756
## converged
## # weights:  106
## initial  value 27.243572
## iter  10 value 4.531543
## iter  20 value 4.453295
## iter  30 value 4.452542
## final  value 4.452541
## converged
## # weights:  22
## initial  value 23.096791
## iter  10 value 3.832659
## iter  20 value 3.807257
## iter  30 value 3.782471
## iter  40 value 3.768660
## iter  50 value 0.098872
## iter  60 value 0.066879
## iter  70 value 0.062846
## iter  80 value 0.051987
## iter  90 value 0.049631
## iter 100 value 0.048572
## final  value 0.048572
## stopped after 100 iterations
## # weights:  64
## initial  value 22.567099
## iter  10 value 3.431509
```

```
## iter  20 value 0.104915
## iter  30 value 0.087769
## iter  40 value 0.049313
## iter  50 value 0.045381
## iter  60 value 0.041540
## iter  70 value 0.038811
## iter  80 value 0.034253
## iter  90 value 0.027691
## iter 100 value 0.025559
## final  value 0.025559
## stopped after 100 iterations
## # weights:  106
## initial  value 23.226649
## iter  10 value 0.966899
## iter  20 value 0.086453
## iter  30 value 0.062040
## iter  40 value 0.036985
## iter  50 value 0.029357
## iter  60 value 0.028210
## iter  70 value 0.026297
## iter  80 value 0.024431
## iter  90 value 0.023005
## iter 100 value 0.022346
## final  value 0.022346
## stopped after 100 iterations
## # weights:  22
## initial  value 22.508972
## iter  10 value 5.652106
## iter  20 value 3.357547
## iter  30 value 3.351054
## iter  40 value 3.351020
## final  value 3.350997
## converged
## # weights:  64
## initial  value 29.956624
## iter  10 value 2.206677
## iter  20 value 0.002944
## final  value 0.000067
## converged
## # weights:  106
## initial  value 27.709377
## iter  10 value 0.030521
## iter  20 value 0.002328
## final  value 0.000060
## converged
## # weights:  22
## initial  value 22.891496
## iter  10 value 8.904338
## iter  20 value 8.850498
## iter  20 value 8.850498
```

```
## iter  20 value 8.850498
## final  value 8.850498
## converged
## # weights:  64
## initial  value 23.776979
## iter  10 value 6.151648
## iter  20 value 5.727553
## iter  30 value 5.726666
## final  value 5.726666
## converged
## # weights:  106
## initial  value 26.431393
## iter  10 value 5.898705
## iter  20 value 5.402260
## iter  30 value 5.401452
## final  value 5.401451
## converged
## # weights:  22
## initial  value 22.736881
## iter  10 value 3.474439
## iter  20 value 3.393031
## iter  30 value 3.390776
## iter  40 value 3.389324
## iter  50 value 3.389094
## iter  60 value 3.389010
## iter  70 value 3.388789
## iter  80 value 3.388674
## final  value 3.388663
## converged
## # weights:  64
## initial  value 26.713071
## iter  10 value 0.473636
## iter  20 value 0.070872
## iter  30 value 0.055912
## iter  40 value 0.049389
## iter  50 value 0.046357
## iter  60 value 0.045329
## iter  70 value 0.044457
## iter  80 value 0.044153
## iter  90 value 0.043930
## iter 100 value 0.043638
## final  value 0.043638
## stopped after 100 iterations
## # weights:  106
## initial  value 22.564649
## iter  10 value 0.454400
## iter  20 value 0.060063
## iter  30 value 0.048467
## iter  40 value 0.040677
## iter  50 value 0.034720
```

```
## iter  60 value 0.031431
## iter  70 value 0.029607
## iter  80 value 0.028361
## iter  90 value 0.027829
## iter 100 value 0.027541
## final  value 0.027541
## stopped after 100 iterations
## # weights:  22
## initial  value 22.712822
## iter  10 value 3.718232
## iter  20 value 0.066865
## iter  30 value 0.000375
## final  value 0.000093
## converged
## # weights:  64
## initial  value 23.087604
## iter  10 value 0.061054
## iter  20 value 0.000525
## final  value 0.000075
## converged
## # weights:  106
## initial  value 22.481784
## iter  10 value 0.275032
## iter  20 value 0.000369
## final  value 0.000095
## converged
## # weights:  22
## initial  value 22.817502
## iter  10 value 7.451991
## iter  20 value 7.117857
## final  value 7.117856
## converged
## # weights:  64
## initial  value 27.059417
## iter  10 value 4.833561
## iter  20 value 4.817221
## final  value 4.817214
## converged
## # weights:  106
## initial  value 25.157137
## iter  10 value 4.522567
## iter  20 value 4.363328
## iter  30 value 4.363200
## iter  30 value 4.363200
## iter  30 value 4.363200
## final  value 4.363200
## converged
## # weights:  22
## initial  value 23.046718
## iter  10 value 0.166474
```

```
## iter  20 value 0.063550
## iter  30 value 0.056704
## iter  40 value 0.049793
## iter  50 value 0.047610
## iter  60 value 0.047539
## iter  70 value 0.047460
## iter  80 value 0.047408
## iter  90 value 0.047381
## iter 100 value 0.047378
## final  value 0.047378
## stopped after 100 iterations
## # weights:  64
## initial  value 22.320004
## iter  10 value 0.185351
## iter  20 value 0.045789
## iter  30 value 0.037534
## iter  40 value 0.034280
## iter  50 value 0.027745
## iter  60 value 0.026052
## iter  70 value 0.023939
## iter  80 value 0.023410
## iter  90 value 0.023180
## iter 100 value 0.023027
## final  value 0.023027
## stopped after 100 iterations
## # weights:  106
## initial  value 22.648572
## iter  10 value 0.144447
## iter  20 value 0.046950
## iter  30 value 0.030020
## iter  40 value 0.027151
## iter  50 value 0.025296
## iter  60 value 0.023778
## iter  70 value 0.022285
## iter  80 value 0.020680
## iter  90 value 0.020305
## iter 100 value 0.019558
## final  value 0.019558
## stopped after 100 iterations
## # weights:  22
## initial  value 23.212294
## iter  10 value 7.481635
## iter  20 value 7.481401
## iter  30 value 7.481003
## iter  40 value 7.479693
## iter  50 value 7.458510
## iter  60 value 3.904525
## iter  70 value 3.804125
## iter  80 value 3.803281
## final  value 3.803207
```

```
## converged
## # weights:  64
## initial  value 20.976067
## iter  10 value 7.449758
## iter  20 value 1.395020
## iter  30 value 1.386309
## final  value 1.386294
## converged
## # weights:  106
## initial  value 25.720986
## iter  10 value 0.349348
## iter  20 value 0.000442
## final  value 0.000065
## converged
## # weights:  22
## initial  value 24.746190
## iter  10 value 9.576190
## final  value 9.574304
## converged
## # weights:  64
## initial  value 25.060724
## iter  10 value 6.304802
## iter  20 value 5.766040
## iter  30 value 5.757962
## final  value 5.757917
## converged
## # weights:  106
## initial  value 28.594572
## iter  10 value 7.038201
## iter  20 value 5.344084
## iter  30 value 5.296910
## iter  40 value 5.296225
## final  value 5.296222
## converged
## # weights:  22
## initial  value 24.055457
## iter  10 value 4.736633
## iter  20 value 3.841656
## iter  30 value 3.840586
## iter  40 value 3.840410
## iter  50 value 3.840268
## iter  60 value 3.840082
## iter  70 value 3.840045
## iter  70 value 3.840045
## iter  70 value 3.840045
## final  value 3.840045
## converged
## # weights:  64
## initial  value 21.436222
## iter  10 value 2.128851
```

```
## iter  20 value 0.103950
## iter  30 value 0.090942
## iter  40 value 0.086192
## iter  50 value 0.079845
## iter  60 value 0.073546
## iter  70 value 0.071186
## iter  80 value 0.069448
## iter  90 value 0.068522
## iter 100 value 0.067826
## final  value 0.067826
## stopped after 100 iterations
## # weights:  106
## initial  value 21.998862
## iter  10 value 0.772065
## iter  20 value 0.087797
## iter  30 value 0.072151
## iter  40 value 0.056059
## iter  50 value 0.044457
## iter  60 value 0.037524
## iter  70 value 0.032360
## iter  80 value 0.031293
## iter  90 value 0.028691
## iter 100 value 0.028088
## final  value 0.028088
## stopped after 100 iterations
## # weights:  22
## initial  value 22.725193
## iter  10 value 5.421190
## iter  20 value 0.096022
## iter  30 value 0.000453
## final  value 0.000057
## converged
## # weights:  64
## initial  value 23.992764
## iter  10 value 0.340285
## iter  20 value 0.000561
## final  value 0.000071
## converged
## # weights:  106
## initial  value 23.177359
## iter  10 value 0.000640
## final  value 0.000063
## converged
## # weights:  22
## initial  value 23.172756
## iter  10 value 7.849011
## iter  20 value 6.204190
## final  value 6.204166
## converged
## # weights:  64
```

```
## initial  value 25.840956
## iter  10 value 5.372676
## iter  20 value 4.131478
## final  value 4.130915
## converged
## # weights:  106
## initial  value 23.839608
## iter  10 value 4.072497
## iter  20 value 3.727671
## iter  30 value 3.605735
## iter  40 value 3.602698
## final  value 3.602696
## converged
## # weights:  22
## initial  value 22.828993
## iter  10 value 5.549149
## iter  20 value 5.451900
## iter  30 value 5.437957
## iter  40 value 5.434106
## iter  50 value 5.430630
## iter  60 value 5.427401
## iter  70 value 5.425744
## iter  80 value 5.425381
## iter  90 value 0.173529
## iter 100 value 0.049125
## final  value 0.049125
## stopped after 100 iterations
## # weights:  64
## initial  value 19.919895
## iter  10 value 0.093340
## iter  20 value 0.079828
## iter  30 value 0.048360
## iter  40 value 0.042816
## iter  50 value 0.034506
## iter  60 value 0.029987
## iter  70 value 0.028201
## iter  80 value 0.022183
## iter  90 value 0.019428
## iter 100 value 0.018721
## final  value 0.018721
## stopped after 100 iterations
## # weights:  106
## initial  value 23.018846
## iter  10 value 0.107496
## iter  20 value 0.070527
## iter  30 value 0.045426
## iter  40 value 0.033579
## iter  50 value 0.027603
## iter  60 value 0.025347
## iter  70 value 0.020334
```

```
## iter  80 value 0.019562
## iter  90 value 0.016585
## iter 100 value 0.015591
## final  value 0.015591
## stopped after 100 iterations
## # weights:  22
## initial  value 20.846298
## iter  10 value 3.891315
## iter  20 value 3.442569
## iter  30 value 3.442032
## iter  30 value 3.442032
## iter  30 value 3.442032
## final  value 3.442032
## converged
## # weights:  64
## initial  value 22.370908
## iter  10 value 0.556436
## iter  20 value 0.001240
## final  value 0.000073
## converged
## # weights:  106
## initial  value 27.407821
## iter  10 value 0.125896
## iter  20 value 0.000713
## final  value 0.000098
## converged
## # weights:  22
## initial  value 22.022959
## iter  10 value 8.531414
## iter  20 value 7.645714
## final  value 7.645711
## converged
## # weights:  64
## initial  value 27.175266
## iter  10 value 5.495263
## iter  20 value 5.298362
## iter  30 value 5.298031
## iter  30 value 5.298031
## iter  30 value 5.298031
## final  value 5.298031
## converged
## # weights:  106
## initial  value 24.151459
## iter  10 value 5.025216
## iter  20 value 4.891467
## iter  30 value 4.890711
## final  value 4.890710
## converged
## # weights:  22
## initial  value 21.765489
```

```
## iter  10 value 3.753595
## iter  20 value 3.478074
## iter  30 value 3.473265
## iter  40 value 3.471135
## iter  50 value 3.470562
## iter  60 value 3.470302
## iter  70 value 3.470106
## iter  80 value 3.469981
## final  value 3.469959
## converged
## # weights:  64
## initial  value 25.447287
## iter  10 value 0.054787
## iter  20 value 0.040965
## iter  30 value 0.038782
## iter  40 value 0.035570
## iter  50 value 0.031167
## iter  60 value 0.029046
## iter  70 value 0.028652
## iter  80 value 0.028244
## iter  90 value 0.028180
## iter 100 value 0.028040
## final  value 0.028040
## stopped after 100 iterations
## # weights:  106
## initial  value 23.450527
## iter  10 value 0.183146
## iter  20 value 0.042077
## iter  30 value 0.034874
## iter  40 value 0.032018
## iter  50 value 0.030179
## iter  60 value 0.027515
## iter  70 value 0.025212
## iter  80 value 0.024327
## iter  90 value 0.023799
## iter 100 value 0.023662
## final  value 0.023662
## stopped after 100 iterations
## # weights:  22
## initial  value 21.660268
## iter  10 value 3.923629
## iter  20 value 3.862158
## iter  30 value 3.862067
## final  value 3.862065
## converged
## # weights:  64
## initial  value 21.790367
## iter  10 value 0.194799
## iter  20 value 0.000463
## final  value 0.000058
```

```
## converged
## # weights:  106
## initial  value 28.735367
## iter  10 value 0.047917
## final  value 0.000065
## converged
## # weights:  22
## initial  value 26.373273
## iter  10 value 7.293349
## iter  20 value 7.221028
## iter  20 value 7.221028
## iter  20 value 7.221028
## final  value 7.221028
## converged
## # weights:  64
## initial  value 23.063974
## iter  10 value 6.686144
## iter  20 value 4.999018
## iter  30 value 4.944183
## iter  40 value 4.942491
## iter  40 value 4.942491
## iter  40 value 4.942491
## final  value 4.942491
## converged
## # weights:  106
## initial  value 29.091379
## iter  10 value 4.482450
## iter  20 value 4.457911
## final  value 4.457870
## converged
## # weights:  22
## initial  value 23.032366
## iter  10 value 4.501118
## iter  20 value 0.068018
## iter  30 value 0.053857
## iter  40 value 0.049645
## iter  50 value 0.048824
## iter  60 value 0.048031
## iter  70 value 0.047905
## iter  80 value 0.047711
## iter  90 value 0.047601
## iter 100 value 0.047574
## final  value 0.047574
## stopped after 100 iterations
## # weights:  64
## initial  value 22.425010
## iter  10 value 0.137925
## iter  20 value 0.112151
## iter  30 value 0.078252
## iter  40 value 0.057639
```

```
## iter  50 value 0.036456
## iter  60 value 0.031054
## iter  70 value 0.027531
## iter  80 value 0.024999
## iter  90 value 0.024034
## iter 100 value 0.023315
## final  value 0.023315
## stopped after 100 iterations
## # weights:  106
## initial  value 27.084576
## iter  10 value 2.078017
## iter  20 value 0.084949
## iter  30 value 0.071149
## iter  40 value 0.061652
## iter  50 value 0.051507
## iter  60 value 0.043948
## iter  70 value 0.036430
## iter  80 value 0.033020
## iter  90 value 0.028389
## iter 100 value 0.024476
## final  value 0.024476
## stopped after 100 iterations
## # weights:  22
## initial  value 21.643997
## iter  10 value 3.930746
## iter  20 value 3.862139
## final  value 3.862065
## converged
## # weights:  64
## initial  value 23.514937
## iter  10 value 3.179466
## iter  20 value 0.013546
## iter  30 value 0.000317
## final  value 0.000084
## converged
## # weights:  106
## initial  value 20.566086
## iter  10 value 0.024704
## iter  20 value 0.000513
## final  value 0.000063
## converged
## # weights:  22
## initial  value 20.355606
## iter  10 value 7.131072
## final  value 7.129987
## converged
## # weights:  64
## initial  value 22.350392
## iter  10 value 5.256678
## iter  20 value 4.838199
```

```
## iter  30 value 4.833844
## final  value 4.833498
## converged
## # weights:  106
## initial  value 26.364939
## iter  10 value 5.231476
## iter  20 value 4.596514
## iter  30 value 4.587229
## final  value 4.587174
## converged
## # weights:  22
## initial  value 23.714249
## iter  10 value 0.159903
## iter  20 value 0.106015
## iter  30 value 0.073775
## iter  40 value 0.070432
## iter  50 value 0.065180
## iter  60 value 0.057275
## iter  70 value 0.050577
## iter  80 value 0.049466
## iter  90 value 0.047205
## iter 100 value 0.047117
## final  value 0.047117
## stopped after 100 iterations
## # weights:  64
## initial  value 23.532120
## iter  10 value 0.051799
## iter  20 value 0.034539
## iter  30 value 0.029486
## iter  40 value 0.025377
## iter  50 value 0.024068
## iter  60 value 0.022938
## iter  70 value 0.022637
## iter  80 value 0.022532
## iter  90 value 0.022506
## iter 100 value 0.022496
## final  value 0.022496
## stopped after 100 iterations
## # weights:  106
## initial  value 24.812266
## iter  10 value 0.279239
## iter  20 value 0.074938
## iter  30 value 0.066091
## iter  40 value 0.049934
## iter  50 value 0.040874
## iter  60 value 0.037546
## iter  70 value 0.028165
## iter  80 value 0.025874
## iter  90 value 0.023165
## iter 100 value 0.021434
```

```
## final  value 0.021434
## stopped after 100 iterations
## # weights:  22
## initial  value 20.400536
## iter  10 value 9.073674
## iter  20 value 9.071060
## iter  30 value 5.281389
## iter  40 value 5.218357
## iter  50 value 5.215534
## final  value 5.215532
## converged
## # weights:  64
## initial  value 32.314929
## iter  10 value 0.308146
## iter  20 value 0.000637
## final  value 0.000086
## converged
## # weights:  106
## initial  value 27.311243
## iter  10 value 0.389918
## iter  20 value 0.000681
## final  value 0.000097
## converged
## # weights:  22
## initial  value 27.583313
## iter  10 value 9.928225
## iter  20 value 7.168596
## final  value 7.168584
## converged
## # weights:  64
## initial  value 23.567782
## iter  10 value 5.940145
## iter  20 value 5.812793
## final  value 5.812629
## converged
## # weights:  106
## initial  value 32.576568
## iter  10 value 6.667136
## iter  20 value 5.123076
## iter  30 value 4.946591
## iter  40 value 4.944534
## final  value 4.944532
## converged
## # weights:  22
## initial  value 21.544242
## iter  10 value 9.087589
## iter  20 value 7.249659
## iter  30 value 0.311299
## iter  40 value 0.070585
## iter  50 value 0.056526
```

```
## iter  60 value 0.049763
## iter  70 value 0.048766
## iter  80 value 0.048364
## iter  90 value 0.048086
## iter 100 value 0.048051
## final  value 0.048051
## stopped after 100 iterations
## # weights:  64
## initial  value 26.672884
## iter  10 value 0.361906
## iter  20 value 0.081781
## iter  30 value 0.066708
## iter  40 value 0.054433
## iter  50 value 0.047224
## iter  60 value 0.045256
## iter  70 value 0.042794
## iter  80 value 0.040934
## iter  90 value 0.039680
## iter 100 value 0.038000
## final  value 0.038000
## stopped after 100 iterations
## # weights:  106
## initial  value 20.782735
## iter  10 value 0.087165
## iter  20 value 0.065130
## iter  30 value 0.053913
## iter  40 value 0.044998
## iter  50 value 0.036062
## iter  60 value 0.033234
## iter  70 value 0.031584
## iter  80 value 0.029730
## iter  90 value 0.028674
## iter 100 value 0.027933
## final  value 0.027933
## stopped after 100 iterations
## # weights:  22
## initial  value 24.624977
## iter  10 value 7.731778
## iter  20 value 3.897971
## iter  30 value 3.862144
## final  value 3.862065
## converged
## # weights:  64
## initial  value 25.382755
## iter  10 value 0.321443
## iter  20 value 0.000448
## final  value 0.000055
## converged
## # weights:  106
## initial  value 25.388034
```

```
## iter  10 value 0.148370
## iter  20 value 0.000208
## final  value 0.000052
## converged
## # weights:  22
## initial  value 23.233713
## iter  10 value 9.762960
## final  value 9.759638
## converged
## # weights:  64
## initial  value 24.349671
## iter  10 value 6.573690
## iter  20 value 5.787501
## iter  30 value 5.784206
## iter  30 value 5.784206
## iter  30 value 5.784206
## final  value 5.784206
## converged
## # weights:  106
## initial  value 23.603575
## iter  10 value 6.190742
## iter  20 value 5.484127
## iter  30 value 5.460790
## iter  40 value 5.460570
## final  value 5.460570
## converged
## # weights:  22
## initial  value 22.175647
## iter  10 value 3.928390
## iter  20 value 3.892503
## iter  30 value 3.891983
## iter  40 value 3.891793
## iter  50 value 3.891725
## iter  60 value 3.891714
## iter  70 value 3.891679
## final  value 3.891651
## converged
## # weights:  64
## initial  value 20.846465
## iter  10 value 2.169339
## iter  20 value 2.000083
## iter  30 value 1.127462
## iter  40 value 0.081371
## iter  50 value 0.063850
## iter  60 value 0.059695
## iter  70 value 0.046614
## iter  80 value 0.042565
## iter  90 value 0.040039
## iter 100 value 0.037007
## final  value 0.037007
```

```
## stopped after 100 iterations
## # weights:  106
## initial  value 22.617479
## iter  10 value 2.708370
## iter  20 value 2.004784
## iter  30 value 1.990735
## iter  40 value 1.979738
## iter  50 value 0.055830
## iter  60 value 0.043501
## iter  70 value 0.040539
## iter  80 value 0.037233
## iter  90 value 0.034189
## iter 100 value 0.031191
## final  value 0.031191
## stopped after 100 iterations
## # weights:  22
## initial  value 25.645339
## iter  10 value 8.110820
## iter  20 value 8.109770
## iter  30 value 0.336366
## iter  40 value 0.001662
## final  value 0.000090
## converged
## # weights:  64
## initial  value 23.399763
## iter  10 value 0.046102
## iter  20 value 0.000570
## final  value 0.000082
## converged
## # weights:  106
## initial  value 24.239063
## iter  10 value 0.121482
## iter  20 value 0.000157
## iter  20 value 0.000079
## iter  20 value 0.000079
## final  value 0.000079
## converged
## # weights:  22
## initial  value 23.460774
## iter  10 value 8.314691
## iter  20 value 7.472479
## iter  30 value 7.471649
## final  value 7.471649
## converged
## # weights:  64
## initial  value 22.768274
## iter  10 value 5.469677
## iter  20 value 5.359851
## final  value 5.359418
## converged
```

```
## # weights:  106
## initial  value 24.726424
## iter  10 value 5.427257
## iter  20 value 5.243033
## iter  30 value 5.233435
## iter  40 value 5.233392
## final  value 5.233391
## converged
## # weights:  22
## initial  value 22.449338
## iter  10 value 8.133787
## iter  20 value 0.128923
## iter  30 value 0.080744
## iter  40 value 0.065093
## iter  50 value 0.051571
## iter  60 value 0.049879
## iter  70 value 0.048006
## iter  80 value 0.047838
## iter  90 value 0.047820
## iter 100 value 0.047809
## final  value 0.047809
## stopped after 100 iterations
## # weights:  64
## initial  value 23.270567
## iter  10 value 0.215324
## iter  20 value 0.101578
## iter  30 value 0.060790
## iter  40 value 0.047363
## iter  50 value 0.043487
## iter  60 value 0.037345
## iter  70 value 0.032353
## iter  80 value 0.031855
## iter  90 value 0.029760
## iter 100 value 0.027901
## final  value 0.027901
## stopped after 100 iterations
## # weights:  106
## initial  value 20.304833
## iter  10 value 0.046999
## iter  20 value 0.033093
## iter  30 value 0.026661
## iter  40 value 0.025233
## iter  50 value 0.023801
## iter  60 value 0.023123
## iter  70 value 0.022645
## iter  80 value 0.022369
## iter  90 value 0.022060
## iter 100 value 0.021531
## final  value 0.021531
## stopped after 100 iterations
```

```
## # weights:  22
## initial  value 22.818702
## iter  10 value 3.848250
## iter  20 value 3.740821
## final  value 3.740667
## converged
## # weights:  64
## initial  value 23.507553
## iter  10 value 0.176506
## iter  20 value 0.000270
## final  value 0.000071
## converged
## # weights:  106
## initial  value 22.967759
## iter  10 value 0.198630
## iter  20 value 0.000245
## final  value 0.000063
## converged
## # weights:  22
## initial  value 21.893820
## iter  10 value 10.642957
## iter  20 value 9.286678
## final  value 9.284037
## converged
## # weights:  64
## initial  value 21.479503
## iter  10 value 6.395938
## iter  20 value 5.552088
## iter  30 value 5.548229
## iter  30 value 5.548229
## iter  30 value 5.548229
## final  value 5.548229
## converged
## # weights:  106
## initial  value 33.413769
## iter  10 value 5.884325
## iter  20 value 4.994144
## iter  30 value 4.986893
## final  value 4.986879
## converged
## # weights:  22
## initial  value 25.752270
## iter  10 value 3.971051
## iter  20 value 3.937705
## iter  30 value 3.921548
## iter  40 value 3.890087
## iter  50 value 3.875790
## iter  60 value 3.851850
## iter  70 value 3.790531
## iter  80 value 3.788292
```

```
## iter  90 value 3.782510
## iter 100 value 3.779073
## final  value 3.779073
## stopped after 100 iterations
## # weights:  64
## initial  value 24.504580
## iter  10 value 4.895257
## iter  20 value 3.793936
## iter  30 value 3.785749
## iter  40 value 3.782095
## iter  50 value 3.659886
## iter  60 value 3.374565
## iter  70 value 2.017881
## iter  80 value 0.755601
## iter  90 value 0.077881
## iter 100 value 0.072293
## final  value 0.072293
## stopped after 100 iterations
## # weights:  106
## initial  value 23.587717
## iter  10 value 0.812003
## iter  20 value 0.050002
## iter  30 value 0.043952
## iter  40 value 0.037752
## iter  50 value 0.033982
## iter  60 value 0.032325
## iter  70 value 0.031581
## iter  80 value 0.031029
## iter  90 value 0.030452
## iter 100 value 0.030023
## final  value 0.030023
## stopped after 100 iterations
## # weights:  22
## initial  value 28.413324
## iter  10 value 4.781048
## iter  20 value 0.050891
## iter  30 value 0.000241
## final  value 0.000065
## converged
## # weights:  64
## initial  value 18.520693
## iter  10 value 7.330340
## iter  20 value 0.045648
## iter  30 value 0.001545
## iter  40 value 0.000167
## iter  50 value 0.000110
## iter  50 value 0.000098
## iter  50 value 0.000098
## final  value 0.000098
## converged
```

```
## # weights:  106
## initial  value 24.374792
## iter  10 value 0.001835
## final  value 0.000082
## converged
## # weights:  22
## initial  value 31.629966
## iter  10 value 12.704964
## iter  20 value 12.675673
## iter  20 value 12.675673
## iter  20 value 12.675673
## final  value 12.675673
## converged
## # weights:  64
## initial  value 24.024678
## iter  10 value 6.469908
## iter  20 value 6.038330
## final  value 6.038211
## converged
## # weights:  106
## initial  value 23.722173
## iter  10 value 5.804935
## iter  20 value 5.445800
## iter  30 value 5.431250
## iter  40 value 5.431228
## iter  40 value 5.431228
## iter  40 value 5.431228
## final  value 5.431228
## converged
## # weights:  22
## initial  value 21.361161
## iter  10 value 7.642754
## iter  20 value 7.305653
## iter  30 value 7.303850
## iter  40 value 7.303642
## iter  50 value 7.303441
## iter  60 value 7.303372
## iter  70 value 7.303316
## iter  70 value 7.303316
## iter  70 value 7.303316
## final  value 7.303316
## converged
## # weights:  64
## initial  value 22.277454
## iter  10 value 0.300196
## iter  20 value 0.154184
## iter  30 value 0.118132
## iter  40 value 0.097272
## iter  50 value 0.082312
## iter  60 value 0.064283
```

```
## iter  70 value 0.053532
## iter  80 value 0.038581
## iter  90 value 0.036243
## iter 100 value 0.035400
## final  value 0.035400
## stopped after 100 iterations
## # weights:  106
## initial  value 23.465670
## iter  10 value 0.185665
## iter  20 value 0.082187
## iter  30 value 0.061120
## iter  40 value 0.047601
## iter  50 value 0.036855
## iter  60 value 0.034078
## iter  70 value 0.030203
## iter  80 value 0.027909
## iter  90 value 0.026735
## iter 100 value 0.026130
## final  value 0.026130
## stopped after 100 iterations
## # weights:  22
## initial  value 21.795759
## iter  10 value 3.354213
## iter  20 value 0.097380
## iter  30 value 0.000436
## final  value 0.000055
## converged
## # weights:  64
## initial  value 21.814273
## iter  10 value 0.267016
## iter  20 value 0.000447
## final  value 0.000058
## converged
## # weights:  106
## initial  value 22.555412
## iter  10 value 0.098589
## iter  20 value 0.000177
## final  value 0.000064
## converged
## # weights:  22
## initial  value 29.573000
## iter  10 value 9.602530
## iter  20 value 9.553258
## final  value 9.553258
## converged
## # weights:  64
## initial  value 23.520581
## iter  10 value 5.647074
## iter  20 value 5.623919
## final  value 5.623911
```

```
## converged
## # weights:  106
## initial  value 28.988729
## iter  10 value 5.286497
## iter  20 value 5.097648
## iter  30 value 5.096952
## final  value 5.096950
## converged
## # weights:  22
## initial  value 21.347174
## iter  10 value 3.739555
## iter  20 value 3.556776
## iter  30 value 3.556146
## iter  40 value 3.555465
## iter  50 value 3.555215
## iter  60 value 3.555160
## iter  70 value 3.555105
## iter  80 value 3.555070
## final  value 3.555024
## converged
## # weights:  64
## initial  value 21.723170
## iter  10 value 0.116882
## iter  20 value 0.071757
## iter  30 value 0.060747
## iter  40 value 0.054443
## iter  50 value 0.047694
## iter  60 value 0.045553
## iter  70 value 0.044620
## iter  80 value 0.044413
## iter  90 value 0.042489
## iter 100 value 0.040275
## final  value 0.040275
## stopped after 100 iterations
## # weights:  106
## initial  value 30.183722
## iter  10 value 0.088731
## iter  20 value 0.046637
## iter  30 value 0.038961
## iter  40 value 0.035555
## iter  50 value 0.031440
## iter  60 value 0.030517
## iter  70 value 0.029097
## iter  80 value 0.028434
## iter  90 value 0.027137
## iter 100 value 0.026075
## final  value 0.026075
## stopped after 100 iterations
## # weights:  22
## initial  value 21.234577
```

```
## iter  10 value 3.451186
## iter  20 value 3.442044
## final  value 3.442032
## converged
## # weights:  64
## initial  value 30.627095
## iter  10 value 0.029788
## iter  20 value 0.008273
## final  value 0.000068
## converged
## # weights:  106
## initial  value 20.789387
## iter  10 value 3.381332
## iter  20 value 1.427534
## iter  30 value 1.386406
## iter  40 value 1.386220
## iter  50 value 0.004498
## final  value 0.000075
## converged
## # weights:  22
## initial  value 21.145975
## iter  10 value 7.798097
## iter  20 value 6.185715
## final  value 6.162730
## converged
## # weights:  64
## initial  value 21.971409
## iter  10 value 4.375148
## iter  20 value 3.894528
## iter  30 value 3.893790
## final  value 3.893789
## converged
## # weights:  106
## initial  value 21.575426
## iter  10 value 3.506949
## iter  20 value 3.388974
## iter  30 value 3.383831
## final  value 3.383826
## converged
## # weights:  22
## initial  value 21.300633
## iter  10 value 3.527602
## iter  20 value 3.491747
## iter  30 value 3.416707
## iter  40 value 0.069677
## iter  50 value 0.049223
## iter  60 value 0.044723
## iter  70 value 0.043357
## iter  80 value 0.043263
## iter  90 value 0.043221
```

```
## iter 100 value 0.043197
## final  value 0.043197
## stopped after 100 iterations
## # weights:  64
## initial  value 25.497408
## iter  10 value 0.083191
## iter  20 value 0.047292
## iter  30 value 0.045032
## iter  40 value 0.038355
## iter  50 value 0.035981
## iter  60 value 0.030384
## iter  70 value 0.026556
## iter  80 value 0.025383
## iter  90 value 0.022304
## iter 100 value 0.021758
## final  value 0.021758
## stopped after 100 iterations
## # weights:  106
## initial  value 26.891574
## iter  10 value 0.034306
## iter  20 value 0.031292
## iter  30 value 0.025142
## iter  40 value 0.023650
## iter  50 value 0.022227
## iter  60 value 0.019067
## iter  70 value 0.017536
## iter  80 value 0.016456
## iter  90 value 0.015118
## iter 100 value 0.014640
## final  value 0.014640
## stopped after 100 iterations
## # weights:  22
## initial  value 25.561089
## iter  10 value 7.832573
## iter  20 value 4.160513
## iter  30 value 3.676480
## iter  40 value 3.674000
## final  value 3.673950
## converged
## # weights:  64
## initial  value 25.291789
## iter  10 value 0.403401
## iter  20 value 0.001369
## final  value 0.000050
## converged
## # weights:  106
## initial  value 27.830864
## iter  10 value 1.322559
## iter  20 value 0.012499
## final  value 0.000098
```

```
## converged
## # weights:  22
## initial  value 23.376480
## iter  10 value 9.287561
## iter  20 value 9.099557
## final  value 9.099556
## converged
## # weights:  64
## initial  value 21.766418
## iter  10 value 7.107333
## iter  20 value 6.637984
## final  value 6.637548
## converged
## # weights:  106
## initial  value 21.048562
## iter  10 value 5.329197
## iter  20 value 5.168698
## iter  30 value 5.121430
## iter  40 value 5.119871
## final  value 5.119871
## converged
## # weights:  22
## initial  value 23.455683
## iter  10 value 0.941662
## iter  20 value 0.054585
## iter  30 value 0.052858
## iter  40 value 0.049357
## iter  50 value 0.048965
## iter  60 value 0.048904
## iter  70 value 0.048742
## iter  80 value 0.048720
## iter  90 value 0.048701
## iter 100 value 0.048699
## final  value 0.048699
## stopped after 100 iterations
## # weights:  64
## initial  value 24.902033
## iter  10 value 0.869036
## iter  20 value 0.061364
## iter  30 value 0.054792
## iter  40 value 0.051271
## iter  50 value 0.047132
## iter  60 value 0.045715
## iter  70 value 0.044038
## iter  80 value 0.043109
## iter  90 value 0.041279
## iter 100 value 0.040437
## final  value 0.040437
## stopped after 100 iterations
## # weights:  106
```

```
## initial  value 21.873915
## iter  10 value 1.738605
## iter  20 value 0.064998
## iter  30 value 0.058047
## iter  40 value 0.050904
## iter  50 value 0.041706
## iter  60 value 0.039098
## iter  70 value 0.037988
## iter  80 value 0.033848
## iter  90 value 0.030754
## iter 100 value 0.027142
## final  value 0.027142
## stopped after 100 iterations
## # weights:  22
## initial  value 21.383212
## final  value 0.000000
## converged
## # weights:  64
## initial  value 20.864388
## iter  10 value 0.006891
## final  value 0.000082
## converged
## # weights:  106
## initial  value 24.123334
## final  value 0.000078
## converged
## # weights:  22
## initial  value 25.797011
## iter  10 value 13.622201
## iter  20 value 6.489962
## iter  30 value 6.487208
## final  value 6.487208
## converged
## # weights:  64
## initial  value 31.378856
## iter  10 value 4.076486
## iter  20 value 3.886861
## iter  30 value 3.886576
## final  value 3.886575
## converged
## # weights:  106
## initial  value 26.004944
## iter  10 value 3.871277
## iter  20 value 3.397147
## final  value 3.397059
## converged
## # weights:  22
## initial  value 23.143842
## iter  10 value 0.070564
## iter  20 value 0.055127
```

```
## iter  30 value 0.051607
## iter  40 value 0.044936
## iter  50 value 0.044232
## iter  60 value 0.043802
## iter  70 value 0.043562
## iter  80 value 0.043543
## iter  90 value 0.043534
## final  value 0.043534
## converged
## # weights:  64
## initial  value 19.714443
## iter  10 value 0.024103
## iter  20 value 0.019544
## iter  30 value 0.018650
## iter  40 value 0.018055
## iter  50 value 0.017929
## iter  60 value 0.017884
## iter  70 value 0.017877
## iter  80 value 0.017875
## iter  90 value 0.017874
## final  value 0.017874
## converged
## # weights:  106
## initial  value 26.224906
## iter  10 value 0.027770
## iter  20 value 0.019060
## iter  30 value 0.016756
## iter  40 value 0.016107
## iter  50 value 0.015835
## iter  60 value 0.014191
## iter  70 value 0.013250
## iter  80 value 0.013126
## iter  90 value 0.013102
## iter 100 value 0.013077
## final  value 0.013077
## stopped after 100 iterations
## # weights:  22
## initial  value 23.271721
## iter  10 value 0.741232
## iter  20 value 0.000823
## final  value 0.000055
## converged
## # weights:  64
## initial  value 19.928265
## iter  10 value 0.041249
## final  value 0.000054
## converged
## # weights:  106
## initial  value 24.632002
## iter  10 value 0.018695
```

```
## final  value 0.000070
## converged
## # weights:  22
## initial  value 22.356131
## iter  10 value 7.388825
## iter  20 value 7.382311
## final  value 7.382311
## converged
## # weights:  64
## initial  value 27.233658
## iter  10 value 5.148929
## iter  20 value 5.116684
## final  value 5.116666
## converged
## # weights:  106
## initial  value 24.819499
## iter  10 value 5.281034
## iter  20 value 4.618744
## iter  30 value 4.612746
## iter  40 value 4.612654
## iter  40 value 4.612654
## iter  40 value 4.612654
## final  value 4.612654
## converged
## # weights:  22
## initial  value 21.546045
## iter  10 value 6.333743
## iter  20 value 6.315712
## iter  30 value 0.058428
## iter  40 value 0.053838
## iter  50 value 0.049243
## iter  60 value 0.048030
## iter  70 value 0.047875
## iter  80 value 0.047752
## iter  90 value 0.047649
## iter 100 value 0.047578
## final  value 0.047578
## stopped after 100 iterations
## # weights:  64
## initial  value 22.258505
## iter  10 value 5.869187
## iter  20 value 1.038714
## iter  30 value 0.091192
## iter  40 value 0.081092
## iter  50 value 0.050881
## iter  60 value 0.044962
## iter  70 value 0.036850
## iter  80 value 0.035839
## iter  90 value 0.032625
## iter 100 value 0.032193
```

```
## final  value 0.032193
## stopped after 100 iterations
## # weights:  106
## initial  value 23.934246
## iter  10 value 0.054680
## iter  20 value 0.034566
## iter  30 value 0.028543
## iter  40 value 0.025338
## iter  50 value 0.021755
## iter  60 value 0.020427
## iter  70 value 0.019669
## iter  80 value 0.019378
## iter  90 value 0.018912
## iter 100 value 0.018847
## final  value 0.018847
## stopped after 100 iterations
## # weights:  22
## initial  value 22.623071
## iter  10 value 3.482834
## iter  20 value 3.442091
## final  value 3.442032
## converged
## # weights:  64
## initial  value 25.063357
## iter  10 value 0.165674
## iter  20 value 0.000222
## final  value 0.000057
## converged
## # weights:  106
## initial  value 25.398504
## iter  10 value 0.105375
## iter  20 value 0.002631
## final  value 0.000030
## converged
## # weights:  22
## initial  value 23.084183
## iter  10 value 7.830909
## iter  20 value 6.885584
## final  value 6.885511
## converged
## # weights:  64
## initial  value 22.217110
## iter  10 value 5.217577
## iter  20 value 4.884489
## final  value 4.884370
## converged
## # weights:  106
## initial  value 33.357037
## iter  10 value 4.456540
## iter  20 value 4.306937
```

```
## iter  30 value 4.305752
## final  value 4.305745
## converged
## # weights:  22
## initial  value 22.709409
## iter  10 value 0.589796
## iter  20 value 0.059671
## iter  30 value 0.049158
## iter  40 value 0.047275
## iter  50 value 0.046970
## iter  60 value 0.046841
## iter  70 value 0.046822
## iter  80 value 0.046801
## iter  90 value 0.046784
## iter 100 value 0.046775
## final  value 0.046775
## stopped after 100 iterations
## # weights:  64
## initial  value 24.824237
## iter  10 value 0.453170
## iter  20 value 0.063859
## iter  30 value 0.051876
## iter  40 value 0.045228
## iter  50 value 0.041084
## iter  60 value 0.040067
## iter  70 value 0.035190
## iter  80 value 0.031950
## iter  90 value 0.031207
## iter 100 value 0.031051
## final  value 0.031051
## stopped after 100 iterations
## # weights:  106
## initial  value 23.123993
## iter  10 value 0.098929
## iter  20 value 0.067138
## iter  30 value 0.040098
## iter  40 value 0.032126
## iter  50 value 0.025921
## iter  60 value 0.024628
## iter  70 value 0.022337
## iter  80 value 0.021302
## iter  90 value 0.020200
## iter 100 value 0.019686
## final  value 0.019686
## stopped after 100 iterations
## # weights:  22
## initial  value 22.097834
## iter  10 value 6.543221
## iter  20 value 6.501716
## final  value 6.501659
```

```
## converged
## # weights:  64
## initial  value 28.927236
## iter  10 value 0.084539
## iter  20 value 0.000341
## final  value 0.000063
## converged
## # weights:  106
## initial  value 21.383145
## iter  10 value 0.025728
## final  value 0.000099
## converged
## # weights:  22
## initial  value 23.553831
## iter  10 value 10.934150
## iter  20 value 6.665923
## iter  30 value 6.660143
## final  value 6.660142
## converged
## # weights:  64
## initial  value 23.275330
## iter  10 value 5.811528
## iter  20 value 5.436712
## iter  30 value 5.429266
## iter  30 value 5.429266
## iter  30 value 5.429266
## final  value 5.429266
## converged
## # weights:  106
## initial  value 24.481208
## iter  10 value 4.537919
## iter  20 value 3.845191
## iter  30 value 3.842966
## final  value 3.842965
## converged
## # weights:  22
## initial  value 22.232076
## iter  10 value 6.593883
## iter  20 value 6.571744
## iter  30 value 6.556088
## iter  40 value 6.535938
## iter  50 value 6.535487
## iter  60 value 6.531983
## iter  70 value 6.529044
## iter  80 value 6.528290
## iter  90 value 6.527610
## iter 100 value 6.527508
## final  value 6.527508
## stopped after 100 iterations
## # weights:  64
```

```
## initial  value 21.464631
## iter  10 value 0.258443
## iter  20 value 0.167109
## iter  30 value 0.048373
## iter  40 value 0.045920
## iter  50 value 0.041404
## iter  60 value 0.039072
## iter  70 value 0.038634
## iter  80 value 0.034400
## iter  90 value 0.032613
## iter 100 value 0.026457
## final  value 0.026457
## stopped after 100 iterations
## # weights:  106
## initial  value 27.571237
## iter  10 value 0.111948
## iter  20 value 0.066587
## iter  30 value 0.050275
## iter  40 value 0.041553
## iter  50 value 0.038713
## iter  60 value 0.033070
## iter  70 value 0.028764
## iter  80 value 0.027484
## iter  90 value 0.023357
## iter 100 value 0.023076
## final  value 0.023076
## stopped after 100 iterations
## # weights:  22
## initial  value 25.402362
## iter  10 value 8.131302
## iter  20 value 7.727447
## final  value 7.727446
## converged
```

```
varImp(modelFit4)
```

```
## nnet variable importance
##
##               Overall
## outer_TGFB1   100.000
## outer_ALPL     70.627
## outer_BMP6     63.266
## outer_BMP2     51.476
## outer_PPARG    50.178
## outer_RunX2    42.184
## outer_OCT4     30.925
## outer_Sox2     21.776
## outer_PTHR     18.909
## outer_Col2A1   14.247
## outer_PDGFB    12.358
## outer_EGFR      9.747
```

```
## outer_VWF       7.224
## outer_Osterix   7.023
## outer_Col1A1    6.695
## outer_IBSP      5.431
## outer_Col10A1   3.608
## outer_NANOG     1.502
## outer_GLA       0.000
```